

# 中华人民共和国通信行业标准

YD/T 2773-2014

---

## 手机支付 基于 2.45GHz RCC(限域通信)技术的 非接触射频接口测试方法

Mobile payment test methods of contactless radio frequency  
interface based on 2.45GHz range controlled communication  
technology

2014-12-24 发布

2014-12-24 实施

---

中华人民共和国工业和信息化部 发布

## 目 次

前 言	II
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 符号和缩略语	3
4.1 符号	3
4.2 缩略语	3
5 物理层测试	3
5.1 测试对象	3
5.2 测试环境	3
5.3 默认容许误差	3
5.4 物理层测试配置	3
5.5 发起方磁通道测试项目	5
5.6 响应方磁通道测试项目	6
6 协议测试	9
6.1 测试对象	9
6.2 测试环境	9
6.3 默认容许误差	9
6.4 协议测试仪	9
6.5 RFU 域	9
6.6 测试命令	9
6.7 协议测试配置	9
6.8 响应方协议测试项目	10
6.9 发起方协议测试项目	37
附录 A (规范性附录) 测试命令	66
附录 B (规范性附录) 标准感应线圈	73
附录 C (资料性附录) 磁通道感应电压与场强的推导	74

## 前 言

本标准按照GB/T 1.1-2009给出的规则起草。

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这些专利的责任。

本标准是手机支付系列标准之一。该系列标准的名称预计如下：

1. 手机支付 术语和定义
2. 手机支付 总体技术要求
3. 手机支付 基于13.56MHz近场通信技术的移动终端技术要求
4. 手机支付 基于13.56MHz近场通信技术的智能卡和内置安全模块技术要求
5. 手机支付 基于2.45GHz RCC（限域通信）技术的智能卡技术要求
6. 手机支付 基于 13.56MHz 近场通信技术的非接触式销售点终端技术要求
7. 手机支付 基于2.45GHz RCC（限域通信）技术的非接触式销售点终端技术要求
8. 手机支付 基于13.56MHz近场通信技术的非接触射频接口技术要求
9. 手机支付 基于2.45GHz RCC（限域通信）技术的非接触射频接口技术要求
10. 手机支付 智能卡和内置模块安全技术要求
11. 手机支付 移动终端安全技术要求
12. 手机支付 多应用管理技术要求
13. 手机支付 基于13.56MHz近场通信技术的移动终端测试方法
14. 手机支付 基于13.56MHz近场通信技术的智能卡和内置安全模块测试方法
15. 手机支付 基于2.45GHz RCC（限域通信）技术的智能卡测试方法
16. 手机支付 基于13.56MHz近场通信技术的非接触式销售点终端测试方法
17. 手机支付 基于2.45GHz RCC（限域通信）技术的非接触式销售点终端测试方法
18. 手机支付 基于13.56MHz 的非接触射频接口测试方法
19. 手机支付 基于2.45GHz RCC（限域通信）技术的非接触射频接口测试方法
20. 手机支付 智能卡和内置模块安全测试方法
21. 手机支付 移动终端安全测试方法
22. 手机支付 多应用管理测试方法

本标准由本标准由中国通信标准化协会提出并归口。

本标准起草单位：工业和信息化部电信研究院、国民技术股份有限公司、中国移动通信集团公司、中国联合通信有限公司、中国电信集团公司。

本标准主要起草人：杨军、袁 琦、吕松栋、杨贤伟、李美祥、黄 鹏、葛 欣、李铭轩、张 强、王兆申、吴 淳、戴军尧、王 逖。



## 手机支付

# 基于 2.45GHz RCC（限域通信）技术的非接触射频接口测试方法

### 1 范围

本标准规定了基于2.45GHz RCC（限域通信）技术的非接触射频接口的测试方法，包括物理层、数据链路与传输会话协议层的测试环境、测试方法步骤等内容。

本标准适用于基于2.45GHz RCC（限域通信）技术的非接触射频接口协议符合性测试。

### 2 规范性引用文件

下列文件对于本标准的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本标准。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本标准。

YD/T 2772 手机支付 基于2.45GHz RCC（限域通信）技术的非接触射频接口技术要求

### 3 术语和定义

下列术语和定义适用于本文件。

#### 3.1

限域通信 Range Controlled Communication  
通信距离范围可控的无线近距离通信技术。

#### 3.2

发起方 Initiator  
2.45GHz手机支付系统距离控制通信的发起方。

#### 3.3

响应方 Target  
2.45GHz手机支付系统对发起方命令请求做出响应的通信方。

#### 3.4

接入标识码 Access Identifier  
用于标识不同的接入响应会话。

#### 3.5

多响应方冲突 Multi Target Collision

多个响应方位于同一个发起方的可接入范围内，发起方将随机地选择任意一个响应方进行接入，使得用户无法直观判断出被接入的响应方，从而造成本次交易具有不确定性。

#### 3.6

冲突检测码 Collision Detect Code  
用于冲突检测的识别码。

#### 3.7

响应方随机标识 Target Random Identifier  
用于冲突检测关闭时，响应方进行连接确认的随机识别码。



3.8

冲突响应时间窗 Collision Response Time Window

响应方在检测到MTC冲突后连续发送冲突响应消息的时间段。

3.9

会话命令超时时间 Session Command Timeout

响应方接收超时：发起方应当在规定的超时时间内发出相应的会话命令，否则响应方认为该次会话发起方超时。

发起方接收超时：响应方应当在规定的超时时间内对发起方的命令做出响应，否则发起方认为该次会话响应方超时。

3.10

磁通道基本消息 Magnetic Channel Message

在磁通道上传输的长度不大于15字节的消息。

3.11

磁通道扩展消息 Extended Magnetic Channel Message

在磁通道上传输的长度大于15字节的消息。

3.12

被测设备 DUT

被测试对象，为响应方或者发起方。

3.13

发起方测试仪 iTester

用于模拟响应方，对发起方DUT进行空中接口协议测试。

3.14

响应方测试仪 tTester

用于模拟发起方，对响应方DUT进行空中接口协议测试。

3.15

测试流程 test scenario

与测试步骤相对应的由一组测试命令及其响应组成的特定序列。

3.16

测试指令 Test Command

测试专用的特殊消息命令。

3.17

工作区域 Operating Range

发起方DUT或者响应方测试仪的工作区域，为响应方在其表面正上方的可接入范围。

3.18

静默 Mute

DUT在接收到测试指令后不做任何响应。

## 4 符号和缩略语

以下符合和缩略语适用于本文件。

### 4.1 符号

bx	数据的第x个bit位，x为十进制数
A(X)xx	‘A’代表具体的测试命令的名称。 ‘X’可以为无效的测试命令‘e’或者有效的测试命令‘t’、‘t1’、‘t2’、‘t3’、‘t4’。 ‘xx’是命令的下标编号，两位十进制数。

### 4.2 缩略语

AID	Access ID	接入标识码
CDC	Collision Detect Code	冲突检测码
DUT	Device Under Test	被测设备
GFSK	Gaussian Frequency-Shift Keying	高斯频移键控
iTester	Initiator Tester	发起方测试仪
LMF	Long Message Format	长消息编码格式
MC	Magnetic Channel	磁通道
MFSSM	Magnetic Field Strength Slope Modulation	磁场场强斜率调制
PRBS	Pseudorandom Binary Sequence	伪随机二进制序列
RC	RF Channel	射频通道
RCC	Range Controlled Communication	限域通信
RF	Radio Frequency	射频
RFU	Reserve for Future Use	保留
tTester	Target Tester	响应方测试仪

## 5 物理层测试

### 5.1 测试对象

本标准针对发起方和响应方分别进行物理层测试，包括对发起方磁通道和射频通道的物理层测试以及对响应方磁通道和射频通道的物理层测试。

### 5.2 测试环境

除非另有说明，测试应在环境温度为 $24^{\circ}\text{C} \pm 3^{\circ}\text{C}$  ( $73^{\circ}\text{F} \pm 5^{\circ}\text{F}$ )、相对湿度为40%至60%的环境下进行。

### 5.3 默认容许误差

除非另有说明，测量设备和测试过程的默认容许误差为 $\pm 5\%$ 。

### 5.4 物理层测试配置

#### 5.4.1 发起方磁通道测试配置

发起方磁通道测试装置包括被测发起方、标准感应线圈、双绞线或同轴电缆、数字示波器及探头，测试配置如图1所示。

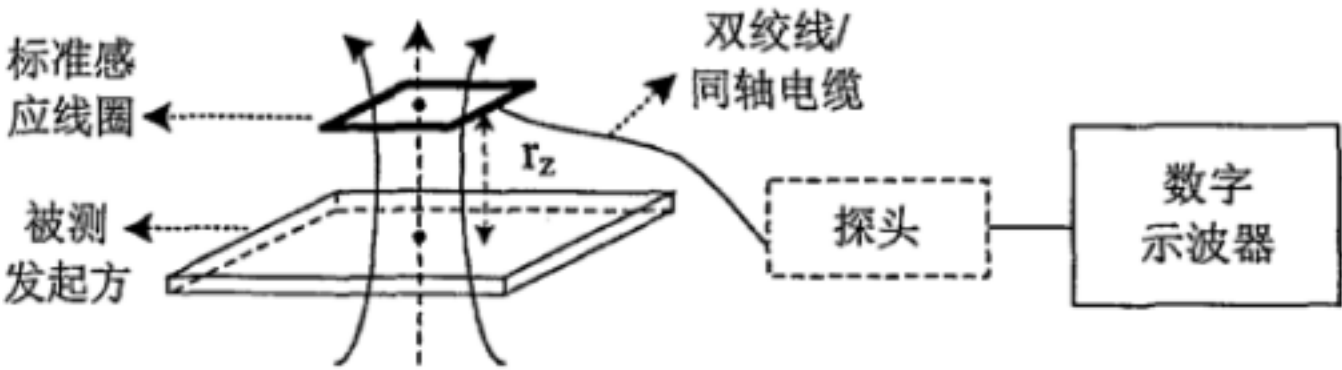


图1 发起方磁通道测试配置示意

- (1) 标准感应线圈符合附录 B 的规定；
- (2) 被测发起方与标准感应线圈相互平行且同轴，其相互之间的垂直距离  $r_z$  在测量不同指标时分别设置如下：
  - 测量被测发起方数据编码符号率和符号周期抖动时， $r_z=0\text{cm}$ ；
  - 测量被测发起方表面磁信号强度时， $r_z=0\text{cm}$ ；
  - 测量被测发起方 10cm 磁信号强度时， $r_z=10\text{cm}$ 。
- (3) 双绞线或者同轴电缆的一端连接标准感应线圈的输出端口，另一端连接数字示波器探头。
- (4) 数字示波器及其探头的带宽应不小于 20kHz，测量精度应不小于 10uV/Div。

5.4.2 响应方磁通道测试配置

响应方磁通道测试装置包括被测响应方、响应方驱动设备以及标准发起方，测试配置如图 2 所示。

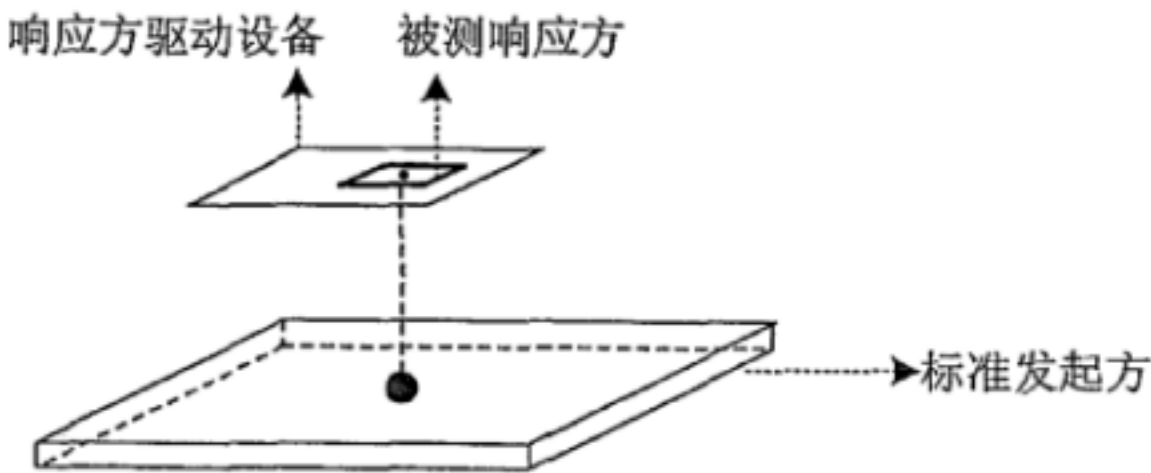


图2 响应方磁通道测试配置示意

- (1) 标准发起方符合《手机支付基于 2.45GHz RCC（限域通信）技术的非接触射频接口技术要求》；
- (2) 响应方驱动设备应能驱动被测响应方正常工作，且对磁通道信号不产生屏蔽作用；
- (3) 被测响应方装载于响应方驱动设备中，被测响应方与标准发起方相互平行且同轴。

5.4.3 发起方/响应方射频通道测试配置

发起方/响应方射频通道测试配置如图 3 所示，DUT 为被测发起方或者被测响应方。

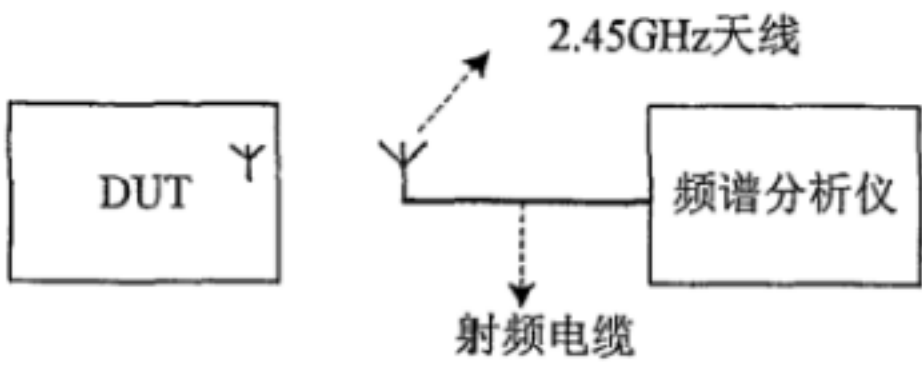


图3 发起方/响应方射频通道测试配置示意

发起方/响应方射频通道测试装置主要指频谱分析仪。频谱分析仪应具备下列功能特性：

- 通用频率域的测量；
- 频率范围为 10MHz 至 3GHz；
- 检波方式为可选峰值或平均值；
- RF 输入阻抗为标称 50Ω；
- 具有 FSK 解调功能。



## 5.5 发起方磁通道测试项目

## 5.5.1 数据编码符号率测试

测试编号: 5.5.1
测试项目: 数据编码符号率
测试目的: 验证发起方磁通道的数据编码符号率是否符合要求
预置条件: 按照图 1 进行连接
测试步骤: 1. 设置被测发起方重复不间断连续发送标准帧格式的 PRBS 数据; 2. 将标准感应线圈置于被测发起方工作方向设备表面垂直距离 0cm 处, 线圈平面与被测发起方设备表面保持平行, 且线圈中心位置与被测发起方设备中心位置重合; 3. 通过标准感应线圈测量被测发起方磁信号的 16 个符号周期总时间为 $t$ ; 4. 根据 $f=16/t$ 计算得到数据编码符号率 $f$
预期结果: 磁通道数据编码符号率为 4kS/s, 符号率容许偏差范围为 $\pm 5\%$

## 5.5.2 磁场信号强度测试

测试编号: 5.5.2
测试项目: 磁场信号强度
测试目的: 验证发起方磁通道的磁场信号强度是否符合要求
预置条件: 按照图1进行连接
测试步骤: 1. 设置被测发起方连续重复不间断地发送 PRBS 数据, 测量其中的(11111111) $b_8$ 数据对应的磁场信号强度; 2. 将标准感应线圈置于被测发起方工作方向设备表面垂直待测距离处, 线圈平面与被测发起方设备表面保持平行, 且线圈中心位置与被测发起方设备中心位置重合; 3. 通过标准感应线圈分别测量被测发起方工作方向设备表面垂直距离 0cm 处和 10cm 处的磁感应电压峰值 $V_{PP}$ , 并由 $H_P=5.4 \times V_{PP}$ 计算相应位置的发起方磁场信号强度
预期结果: 以被测发起方设备工作位置中心为基准, 工作方向设备表面垂直距离 0cm 处磁场信号强度峰值应不小于 160A/m 且不大于 300A/m; 设备表面垂直距离 10cm 处磁场信号强度峰值应小于 4.2A/m

## 5.5.3 磁场信号符号周期抖动测试

测试编号: 5.5.3
测试项目: 磁场信号符号周期抖动
测试目的: 验证发起方磁通道的磁场信号符号周期抖动是否符合要求
预置条件: 按照图1进行连接
测试步骤: 1. 标准感应线圈测量位置位于被测发起方工作方向设备表面垂直距离 0cm 处, 线圈平面与设备表面保持平行, 且中心位置与被测发起方设备中心位置重合; 2. 设置被测发起方连续不间断发送标准帧格式的 PRBS 数据; 3. 通过示波器采集标准感应线圈感应的磁场信号, 并设置示波器为“上升/下降时间”触发模式, 读取触发点后第三个过零点的时间抖动
预期结果: 磁场信号符号周期抖动应不大于 60 $\mu$ s

## 5.6 响应方磁通道测试项目

## 5.6.1 磁场信号强度门限测试

测试编号: 5.6.1
测试项目: 磁场信号强度门限
测试目的: 验证响应方的磁场信号强度门限是否符合要求
预置条件: 按照图 2 进行连接
测试步骤: 1. 设置标准发起方连续不间断发送标准帧格式的 PRBS 数据; 2. 将标准感应线圈置于标准发起方工作方向区域, 线圈平面与标准发起方设备表面保持平行, 且线圈中心位置与标准发起方设备中心位置重合; 3. 通过标准感应线圈确定被测磁场信号强度值所对应的标准发起方工作方向设备表面垂直距离; 4. 将被测响应方置于步骤 3 所确定的距离位置处, 并测试被测响应方是否能够与标准发起方建立并保持连接; 5. 重复步骤 3 到 4, 分别测试在磁场信号强度为 6.7A/m 和 4.2A/m 处被测响应方的连接状态, 应分别符合技术要求
预期结果: 被测响应方设备置于标准发起方设备的磁场信号工作区域, 在标准发起方磁场信号强度峰值为6.7A/m处, 被测响应方应能够与标准发起方建立并保持连接; 被测响应方设备置于标准发起方设备的磁场信号工作区域, 在标准发起方磁场信号强度峰值为4.2A/m处, 响应方应不能够与发起方建立或保持连接

## 5.7 发起方/响应方射频通道测试项目

## 5.7.1 发射功率测试

测试编号: 5.7.1
测试项目: 发射功率
测试目的: 验证发起方或响应方的射频发射功率是否符合要求
预置条件: 1. 按照图 3 进行连接; 2. 在屏蔽室环境下进行测试
测试步骤: 1. 将 2.45GHz 天线通过射频电缆连接到频谱分析仪; 2. 设置被测设备以最大功率通过射频通道发送 PRBS 数据; 3. 测试射频信号峰值功率; 4. 重复步骤 2 到 3, 对中心频率分别为 2468MHz、2434MHz、2401MHz 的 3 个信道进行测试
预期结果: 各射频信道发射功率最大不得超过+3dBm

## 5.7.2 频率容限测试

测试编号: 5.7.2
测试项目: 频率容限
测试目的: 验证发起方或响应方的射频频率容限是否符合要求
预置条件: 1. 按照图 3 进行连接; 2. 在屏蔽室环境下进行测试
测试步骤: 1. 将 2.45GHz 天线通过射频电缆连接到频谱分析仪; 2. 设置被测设备通过射频通道重复发送 0x55 数据; 3. 通过频谱分析仪测量射频通道信号的中心频率 $F_t$ , 并测量 $F_t$ 在一个数据帧传输时间内的累计漂移量; 4. 重复步骤 3, 对中心频率分别为 2468MHz、2434MHz、2401MHz 的 3 个信道进行测试
预期结果: 发射载波的初始中心频率 $F_t$ 应在本信道标称中心频率 $F_c$ 的 $\pm 75\text{kHz}$ 频率范围内, 即: $F_c - 75\text{ kHz} \leq F_t \leq F_c + 75\text{ kHz}.$ 注: $\pm 75\text{kHz}$ 不包括数据发送过程中的频率漂移。 在一个数据帧传输时间内, 发射载波中心频率累积漂移量应在 $\pm 20\text{kHz}$ 之内



5.7.3 调制参数测试

测试编号: 5.7.3
测试项目: 调制参数
测试目的: 验证发起方或响应方的射频调制参数是否符合要求
预置条件: 按照图 3 进行连接; 2、在屏蔽室环境下进行测试
测试步骤: 调制参数测试方法为: 1. 将 2.45GHz 天线通过射频电缆连接到频谱分析仪; 2. 设置被测设备通过射频通道重复发送 0x0F 数据; 3. 通过频谱分析仪测量数据序列传输中的频率偏移幅度为 $F_d$ ; 4. 设置被测设备通过射频通道重复发送 0xAA 数据; 5. 通过频谱分析仪测量数据序列传输中的频率偏移幅度为 $F_{min}$ ; 6. 重复步骤 2 到 5, 对中心频率分别为 2468MHz, 2434MHz, 2401MHz 的 3 个信道进行测试
预期结果: 1. 频率偏移幅度应在 135kHz 至 275kHz。 2. 在 1010 数据序列传输中对应的最小频率偏移幅度 $F_{min}$ 应当不小于 ±80% 的 00001111 数据序列传输中的频率偏移幅度 $F_d$ 。在任何情况下, 最小频率偏移均不应小于 115kHz

5.7.4 带内杂散测试

测试编号：5.7.4						
测试项目：带内杂散						
测试目的： 验证发起方或响应方的射频带内杂散是否符合要求						
预置条件： 1. 按照图 3 进行连接； 2. 在屏蔽室环境下进行测试						
测试步骤： 1. 将 2.45GHz 天线通过射频电缆接到频谱分析仪； 2. 设置被测设备通过被测信道即第 $M$ 信道发送 PRBS 数据； 3. 通过频谱分析仪中心频率测量相邻第 $N$ 信道的功率值总和为相邻第 $N$ 信道上的杂散功率； 4. 重复步骤 3，完成所有相邻信道的测试； 5. 重复步骤 3 到 4，对中心频率分别为 2468MHz、2434MHz、2401MHz 的 3 个信道进行测试						
预期结果： 当射频信号发射设备在第 $M$ 信道发射包含伪随机二进制序列（PRBS）数据的射频信号时，相邻第 $N$ 信道上的杂散功率应小于如下表所示的最大限制。 发射频谱带内杂散						
<table><tr><td>相邻信道间隔</td><td>杂散功率最大限值</td></tr><tr><td>2MHz(<math> M-N =2</math>)</td><td>-20dBm</td></tr><tr><td><math>\geq 3</math>MHz(<math> M-N \geq 3</math>)</td><td>-30dBm</td></tr></table>	相邻信道间隔	杂散功率最大限值	2MHz( $ M-N =2$ )	-20dBm	$\geq 3$ MHz( $ M-N \geq 3$ )	-30dBm
相邻信道间隔	杂散功率最大限值					
2MHz( $ M-N =2$ )	-20dBm					
$\geq 3$ MHz( $ M-N \geq 3$ )	-30dBm					

## 6 协议测试

### 6.1 测试对象

本标准针对发起方和响应方分别进行协议测试。

### 6.2 测试环境

除非另有说明，测试应在常温、相对湿度为40%至60%的环境下进行。

### 6.3 默认容许误差

除非另有说明，测试过程中空口交互协议的时间指标默认容许误差为±30%。

### 6.4 协议测试仪

本标准使用响应方测试仪tTester和发起方测试仪iTester两种测试装置，分别用于响应方DUT和发起方DUT的测试。

iTester和tTester应能测量两次I/O之间的时间，并能记录和检查来自DUT的数据。

iTester和tTester的空中接口物理层和链路层必须符合YD/T XXX.10。

iTester和tTester应能够按照YD/T XXX.10的要求产生I/O字符流。

### 6.5 RFU 域

如果RFU域未设置为其默认值，则测试失败，且认为DUT不符合规定。

### 6.6 测试命令

本标准定义的测试命令参见附录A。

DUT必须支持ECHO数据交易指令APDATA\_REQ(t1)、APDATA\_REQ(t2)。

### 6.7 协议测试配置

#### 6.7.1 响应方协议测试配置

使用响应方测试仪tTester执行测试命令，接收并检查DUT的响应。测试配置如图4所示。

注意：在进行各项测试之前务必将DUT置于响应方测试仪tTester的工作区域。

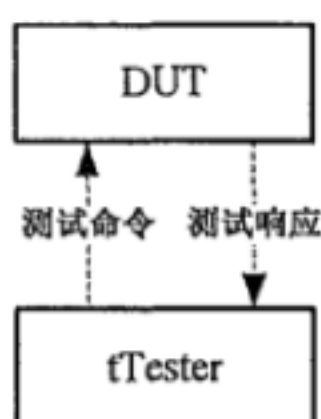


图4 响应方测试配置示意

#### 6.7.2 发起方协议测试配置

使用发起方测试仪iTester通过串口线与DUT连接，并控制DUT执行测试命令，iTester模拟响应方接收并检查DUT的请求。测试配置如图5所示。

注意：在进行各项测试之前务必将发起方测试仪iTester置于DUT的工作区域。

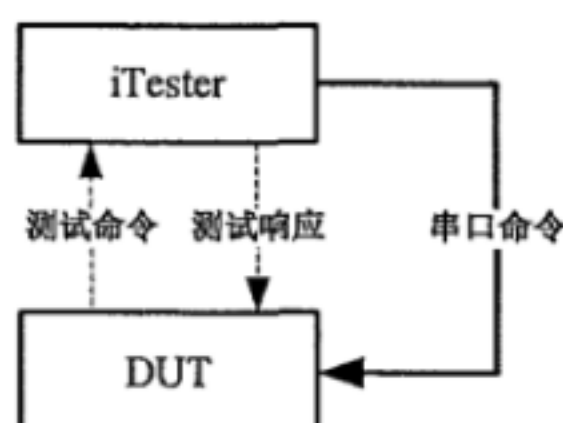


图5 发起方测试配置示意

## 6.8 响应方协议测试项目

## 6.8.1 激活测试

测试编号：6.8.1	
测试项目：激活测试	
测试目的： 验证响应方DUT在激活阶段的数据格式、逻辑操作和响应时间是否符合要求	
预置条件：按照图4进行连接	
测试步骤： 测试 DUT 收到无效的激活命令和正确的激活命令时响应是否正确。 1. tTester发送无效的激活命令INQUIRY(e) <sub>00</sub> ； 2. tTester在8ms内接收DUT的响应； 3. tTester发送有效的激活命令INQUIRY(t) <sub>00</sub> ； 4. tTester在8ms内接收DUT的响应ATI <sub>00</sub> ； 5. tTester检查ATI <sub>00</sub> 数据格式。	
测试流程	
响应方测试仪tTester	DUT
INQUIRY(e) <sub>00</sub>	Mute
INQUIRY(t) <sub>00</sub>	ATI <sub>00</sub>
预期结果： 1. 执行上述测试步骤，DUT的响应与测试流程一致。 2. ATI <sub>00</sub> 的数据格式与下表一致	
字段	期望值
Rfu (4 bit)	0
FormatType(4 bit)	8
Status(8 bit)	0x00或者0xD0h~0xFFh
MsgCode(8 bit)	16
MsgLen(16 bit)	24
IDs(5 byte)	随机数
AccessVersion (1 byte)	0x03
Mac(4byte)	响应方使用 K0 作为密钥对（IDs  TargetID  AccessVersion）进行 MAC 计算。16 字节 K0 使用 SKG0 中同样的扩展方式由 14 字节 IDm 扩展生成。MAC算法见YD/T XXX.10的附录B.5
Reserved (6Byte)	0
CheckSum( 2 Byte)	对消息头和消息体数据进行CheckSum校验



6.8.2 连接测试

测试编号：6.8.2	
测试项目：连接测试	
测试目的： 验证响应方DUT在建立连接阶段的数据格式、逻辑操作和响应时间是否符合要求	
预置条件：按照图4进行连接	
测试步骤： 测试步骤 1： 测试 DUT 在连接阶段收到数据交换请求时响应是否正确。 1. tTester发送激活命令INQUIRY(t) <sub>00</sub> ； 2. tTester在8ms内接收DUT的激活响应ATI <sub>00</sub> ； 3. tTester发送APDATA_REQ(t1) <sub>00</sub> ； 4. tTester在500ms内接收DUT的响应。	
测试流程 1	
响应方测试仪tTester	DUT
INQUIRY(t) <sub>00</sub>	ATI <sub>00</sub>
APDATA_REQ(t1) <sub>00</sub>	Mute
测试步骤 2： 测试 DUT 在连接阶段收到维持连接请求时响应是否正确。 1. tTester发送激活命令INQUIRY(t) <sub>01</sub> ； 2. tTester在8ms内接收DUT的激活响应ATI <sub>01</sub> ； 3. tTester的发送LINKCTL_REQ(t) <sub>01</sub> ； 4. tTester在8ms内接收DUT的响应。	
测试流程 2	
响应方测试仪tTester	DUT
INQUIRY(t) <sub>01</sub>	ATI <sub>01</sub>
LINKCTL_REQ(t) <sub>01</sub>	Mute
测试步骤 3： 测试 DUT 在连接阶段收到关闭连接请求时响应是否正确。 1. tTester发送激活命令INQUIRY(t) <sub>02</sub> ； 2. tTester在8ms内接收DUT的激活响应ATI <sub>02</sub> ； 3. tTester发送CLOSE_REQ(t1) <sub>02</sub> ； 4. tTester在500ms内接收DUT的响应。	

测试流程 3	
响应方测试仪tTester	DUT
INQUIRY(t) <sub>02</sub>	ATI <sub>02</sub>
CLOSE_REQ(t1) <sub>02</sub>	Mute
测试步骤 4:	
测试 DUT 在连接阶段收到无效的连接请求时响应是否正确。	
1. tTester发送激活命令INQUIRY(t) <sub>03</sub> ;	
2. tTester在8ms内接收DUT的激活响应ATI <sub>03</sub> ;	
3. tTester发送无效的连接命令CONNECT_REQ(e) <sub>03</sub> ;	
4. tTester在8ms内接收DUT的响应。	
测试流程 4	
响应方测试仪tTester	DUT
INQUIRY(t) <sub>03</sub>	ATI <sub>03</sub>
CONNECT_REQ(e) <sub>03</sub>	Mute
测试步骤 5:	
测试 DUT 在连接阶段收到有效的连接请求时响应是否正确。	
1. tTester发送激活命令INQUIRY(t) <sub>04</sub> ;	
2. tTester在8ms内接收DUT的激活响应ATI <sub>04</sub> ;	
3. tTester发送有效的连接命令CONNECT_REQ(t) <sub>04</sub> ;	
4. tTester在8ms内接收DUT的连接响应CONNECT_RSP <sub>04</sub> , tTester检查CONNECT_RSP <sub>04</sub> 的数据格式。	
测试流程 5	
响应方测试仪tTester	DUT
INQUIRY(t) <sub>04</sub>	ATI <sub>04</sub>
CONNECT_REQ(t) <sub>04</sub>	CONNECT_RSP <sub>04</sub>
预期结果:	
1. 执行测试步骤1~5, DUT按照测试流程所述进行响应。	
2. CONNECT_RSP <sub>04</sub> 数据格式符合下表要求:	
字段	值
Rfu (4 bit)	0
FormatType(4 bit)	8
Status(8 bit)	0x00或者0xD0h~0xFFh

MsgCode(8 bit)	18
MsgLen(16 bit)	24
Result(1 Byte)	0x00或0x01
RootKeyIndex(1 Byte)	0x00
SessionKey(1 Byte)	0x01
EncAlg(2 Byte)	0x0100
SDInfo(5 Byte)	自定义
SDRand (8 Byte)	随机数
Reseved(6 Byte)	0
Checksum(2 Byte)	对消息头和消息体数据进行Checksum校验

注意：除此测试项之外，所有测试项 DUT 返回的 CONNECT\_RSP 的 ‘Result’ 必须为 0x00



6.8.3 数据交换测试

测试编号：6.8.3												
测试项目：数据交换测试												
测试目的： 验证响应方DUT在数据交易阶段的数据格式、逻辑操作和响应时间是否符合要求												
预置条件：按照图4进行连接												
测试步骤： 测试步骤 1： 测试 DUT 在数据交易阶段接收到无效的数据交换命令时响应是否正确。 1. tTester发送激活命令INQUIRY(t) <sub>00</sub> ； 2. tTester在8ms内接收DUT的激活响应ATI <sub>00</sub> ； 3. tTester发送有效的连接命令CONNECT_REQ(t) <sub>00</sub> ； 4. tTester在8ms内接收DUT的连接响应CONNECT_RSP <sub>00</sub> ； 5. tTester在MC通道持续发送CHECK1_REQ(t) <sub>00</sub> 或者CHECK2_REQ(t) <sub>00</sub> ； 6. tTester发送无效的数据交换命令APDATA_REQ(e) <sub>00</sub> ； 7. tTester在500ms内接收DUT的响应； 8. tTester发送数据交换命令APDATA_REQ(t1) <sub>00</sub> ； 9. tTester在500ms内接收DUT的响应。												
测试流程 1												
<table><tr><td>响应方测试仪tTester</td><td>DUT</td></tr><tr><td>INQUIRY(t)<sub>00</sub></td><td>ATI<sub>00</sub></td></tr><tr><td>CONNECT_REQ(t)<sub>00</sub></td><td>CONNECT_RSP<sub>00</sub></td></tr><tr><td>CHECK1_REQ(t)<sub>00</sub> or CHECK2_REQ(t)<sub>00</sub></td><td></td></tr><tr><td>APDATA_REQ(e)<sub>00</sub></td><td>Mute</td></tr><tr><td>APDATA_REQ(t1)<sub>00</sub></td><td>Mute</td></tr></table>	响应方测试仪tTester	DUT	INQUIRY(t) <sub>00</sub>	ATI <sub>00</sub>	CONNECT_REQ(t) <sub>00</sub>	CONNECT_RSP <sub>00</sub>	CHECK1_REQ(t) <sub>00</sub> or CHECK2_REQ(t) <sub>00</sub>		APDATA_REQ(e) <sub>00</sub>	Mute	APDATA_REQ(t1) <sub>00</sub>	Mute
响应方测试仪tTester	DUT											
INQUIRY(t) <sub>00</sub>	ATI <sub>00</sub>											
CONNECT_REQ(t) <sub>00</sub>	CONNECT_RSP <sub>00</sub>											
CHECK1_REQ(t) <sub>00</sub> or CHECK2_REQ(t) <sub>00</sub>												
APDATA_REQ(e) <sub>00</sub>	Mute											
APDATA_REQ(t1) <sub>00</sub>	Mute											
测试步骤 2： 测试 DUT 在数据交易阶段接收到数据交换命令（单包）时响应是否正确。 1. tTester发送激活命令INQUIRY(t) <sub>01</sub> ； 2. tTester在8ms内接收DUT的激活响应ATI <sub>01</sub> ； 3. tTester发送有效的连接命令CONNECT_REQ(t) <sub>01</sub> ； 4. tTester在8ms内接收DUT的连接响应CONNECT_RSP <sub>01</sub> ； 5. tTester在MC通道持续发送CHECK1_REQ(t) <sub>01</sub> 或者CHECK2_REQ(t) <sub>01</sub> ；												

- 6. tTester发送数据交换命令APDATA\_REQ(t1)<sub>01</sub> ;
- 7. tTester在500ms内接收DUT的数据交换响应APDATA\_RSP(t)<sub>01</sub> ;
- 8. 记录并检查APDATA\_RSP<sub>01</sub>的数据格式，之后延时等待至DUT返回到激活阶段。

测试流程 2

响应方测试仪iTester		DUT
INQUIRY(t) <sub>01</sub>	————→	
	←————	ATI <sub>01</sub>
CONNECT_REQ(t) <sub>01</sub>	————→	
	←————	CONNECT_RSP <sub>01</sub>
CHECK1_REQ(t) <sub>01</sub> or CHECK2_REQ(t) <sub>01</sub>	————→	
APDATA_REQ(t1) <sub>01</sub>	————→	
	←————	APDATA_RSP <sub>01</sub>

测试步骤 3:

测试 DUT 在数据交易阶段接收到数据交换命令（多包）时响应是否正确。

- 1. tTester发送激活命令INQUIRY(t)<sub>02</sub> ;
- 2. tTester在8ms内接收DUT的激活响应ATI<sub>02</sub> ;
- 3. tTester发送有效的连接命令CONNECT\_REQ(t)<sub>02</sub> ;
- 4. tTester在8ms内接收DUT的连接响应CONNECT\_RSP<sub>02</sub> ;
- 5. tTester在MC通道持续发送CHECK1\_REQ(t)<sub>02</sub>或者CHECK2\_REQ(t)<sub>02</sub> ;
- 6. tTester发送数据交换命令APDATA\_REQ(t2)<sub>02</sub> ;
- 7. tTester在500ms内接收DUT的数据交换响应APDATA\_RSP<sub>02</sub>，记录并检查APDATA\_RSP<sub>02</sub>的数据格式。

测试流程 3

响应方测试仪tTester		DUT
INQUIRY(t) <sub>02</sub>	————→	
	←————	ATI <sub>02</sub>
CONNECT_REQ(t) <sub>02</sub>	————→	
	←————	CONNECT_RSP <sub>02</sub>
CHECK1_REQ(t) <sub>02</sub> or CHECK2_REQ(t) <sub>02</sub>	————→	
APDATA_REQ(t2) <sub>02</sub>	————→	
	←————	APDATA_RSP <sub>02</sub>

测试步骤 4:

测试 DUT 在数据交易阶段接收到长度超出 YD/T XXX.10 范围的数据交换命令时响应是否正确。

- 1. tTester发送激活命令INQUIRY(t)<sub>03</sub> ;
- 2. tTester在8ms内接收DUT的激活响应ATI<sub>03</sub> ;
- 3. tTester发送有效的连接命令CONNECT\_REQ(t)<sub>03</sub> ;
- 4. tTester在8ms内接收DUT的连接响应CONNECT\_RSP<sub>03</sub> ;

- 5. tTester在MC通道持续发送CHECK1\_REQ(t)<sub>03</sub>或者CHECK2\_REQ(t)<sub>03</sub>；
- 6. tTester发送数据交换命令APDATA\_REQ(t3)<sub>03</sub>；
- 7. tTester在500ms内接收DUT的响应；
- 8. tTester发送数据交换命令APDATA\_REQ(t1)<sub>03</sub>；
- 9. tTester在500ms内接收DUT的响应。

测试流程 4

响应方测试仪tTester		DUT
INQUIRY(t) <sub>03</sub>	————→	
	←————	ATI <sub>03</sub>
CONNECT_REQ(t) <sub>03</sub>	————→	
	←————	CONNECT_RSP <sub>03</sub>
CHECK1_REQ(t) <sub>03</sub> or CHECK2_REQ(t) <sub>03</sub>	————→	
APDATA_REQ(t3) <sub>03</sub>	————→	
	←————	Mute
APDATA_REQ(t1) <sub>03</sub>	————→	
	←————	Mute

测试步骤 5:

测试 DUT 在数据交易阶段接收到“CDC”错误的冲突检测命令 CHECK1\_REQ(e)时是否正确地更新消息的状态“Status”。

- 1. tTester发送激活命令INQUIRY(t)<sub>04</sub>；
- 2. tTester在8ms内接收DUT的激活响应ATI<sub>04</sub>；
- 3. tTester发送有效的连接命令CONNECT\_REQ(t)<sub>04</sub>；
- 4. tTester在8ms内接收DUT的连接响应CONNECT\_RSP<sub>04</sub>；
- 5. tTester发送CHECK1\_REQ(e)<sub>04</sub>；
- 6. tTester发送APDATA\_REQ(t1)<sub>04</sub>；
- 7. tTester在500ms内接受APDATA\_RSP<sub>04</sub>，检查其Status字段的值。

测试流程 5

响应方测试仪iTester		DUT
INQUIRY(t) <sub>04</sub>	————→	
	←————	ATI <sub>04</sub>
CONNECT_REQ(t) <sub>04</sub>	————→	
	←————	CONNECT_RSP <sub>04</sub>
CHECK1_REQ(e) <sub>04</sub>	————→	
APDATA_REQ(t) <sub>04</sub>	————→	
	←————	APDATA_RSP <sub>04</sub>

测试步骤 6:

测试 DUT 在数据交易阶段接收到“CDC”错误的连接确认消息 CHECK2\_REQ(e)时是否正确地更新消息的状态“Status”。



- 1. tTester发送激活命令INQUIRY(t)<sub>05</sub> ;
- 2. tTester在8ms内接收DUT的激活响应ATI<sub>05</sub> ;
- 3. tTester发送有效的连接命令CONNECT\_REQ(t)<sub>05</sub> ;
- 4. tTester在8ms内接收DUT的连接响应CONNECT\_RSP<sub>05</sub> ;
- 5. tTester发送CHECK2\_REQ(e)<sub>05</sub> ;
- 6. tTester发送APDATA\_REQ(t1)<sub>05</sub> ;
- 7. tTester在500ms内接受APDATA\_RSP<sub>05</sub>, 检查其Status字段的值。

测试流程 6

响应方测试仪iTester		DUT
INQUIRY(t) <sub>05</sub>	————→	
	←————	ATI <sub>05</sub>
CONNECT_REQ(t) <sub>05</sub>	————→	
	←————	CONNECT_RSP <sub>05</sub>
CHECK2_REQ(e) <sub>04</sub>	————→	
APDATA_REQ(t1) <sub>05</sub>	————→	
	←————	APDATA_RSP <sub>05</sub>

测试步骤 7:

测试 DUT 在数据交易阶段收到处理时间超过 500ms 的数据交换指令时, 是否正确地响应 LTW。

- 1. tTester发送激活命令INQUIRY(t)<sub>06</sub> ;
- 2. tTester在8ms内接收DUT的激活响应ATI<sub>06</sub> ;
- 3. tTester发送有效的连接命令CONNECT\_REQ(t)<sub>06</sub> ;
- 4. tTester在8ms内接收DUT的连接响应CONNECT\_RSP<sub>06</sub> ;
- 5. tTester在MC通道持续发送CHECK1\_REQ(t)<sub>06</sub>或者CHECK2\_REQ(t)<sub>06</sub> ;
- 6. tTester发送数据交换命令APDATA\_REQ(t4)<sub>06</sub> ;
- 7. tTester在500ms内接收LTW<sub>06</sub>命令, 重复此步骤直至接收到APDATA\_RSP<sub>06</sub> ;
- 8. tTester记录并检查LTW<sub>06</sub>数据格式和APDATA\_RSP<sub>06</sub>的数据格式。

测试流程 7

响应方测试仪tTester		DUT
INQUIRY(t) <sub>06</sub>	————→	
	←————	ATI <sub>06</sub>
CONNECT_REQ(t) <sub>06</sub>	————→	
	←————	CONNECT_RSP <sub>06</sub>
CHECK1_REQ(t) <sub>06</sub> or CHECK2_REQ(t) <sub>06</sub>	————→	
APDATA_REQ(t4) <sub>06</sub>	————→	
	←————	LTW <sub>06</sub>
	.....	.....
	←————	LTW <sub>06</sub>
	←————	APDATA_RSP <sub>06</sub>

预期结果:

1. 执行上述测试步骤1~7, DUT按照测试流程1~7所述进行响应。
2. APDATA\_RSP<sub>04</sub>和APDATA\_RSP<sub>05</sub>的Status字段值为0x01。
3. APDATA\_RSP<sub>01</sub>的数据格式符合下表要求:

字段	值
Rfu (4 bit)	0
FormatType(4 bit)	8
Status(8 bit)	0x00或者0xD0h~0xFFh
MsgCode(8 bit)	20
MsgLen(16 bit)	24
EncPayLoad(24 Byte)	与APDATA_REQ(t1) <sub>01</sub> 中EncPayLoad匹配, 详见附录A
Checksum(2 Byte)	对消息头和消息体数据进行Checksum校验

4. APDATA\_RSP<sub>02</sub>的数据格式符合下表要求:

字段	值
Rfu (4 bit)	0
FormatType(4 bit)	8
Status(8 bit)	除0x01、0x02、0x82的其他数值
MsgCode(8 bit)	20
MsgLen(16 bit)	248
EncPayLoad(248 Byte)	与APDATA_REQ(t2) <sub>02</sub> 中EncPayLoad相同
Checksum(2 Byte)	对消息头和消息体数据进行Checksum校验

5. APDATA\_RSP<sub>06</sub>的数据格式符合下表要求:

字段	值
Rfu (4 bit)	0
FormatType(4 bit)	8
Status(8 bit)	0x00或者0xD0h~0xFFh
MsgCode(8 bit)	20
MsgLen(16 bit)	自定义, 与EncPayLoad匹配, 不超过288
EncPayLoad(小于288 Byte)	自定义, 与DUT匹配
Checksum(2 Byte)	对消息头和消息体数据进行Checksum校验

6. LTW<sub>06</sub>的数据格式符合下表要求:

字段	值
Rfu (4 bit)	0
FormatType(4 bit)	8
Status(8 bit)	0x00或者0xD0h~0xFFh
MsgCode(8 bit)	25
MsgLen(16 bit)	2
RandData(1Byte)	随机数
Reserved(1 Byte)	0
Checksum(2 Byte)	对消息头和消息体数据进行Checksum校验

6.8.4 链路维持测试

测试编号：6.8.4	
测试项目：链路维持测试	
测试目的： 验证响应方DUT在链路维持阶段的数据格式、逻辑操作和响应时间是否符合要求	
预置条件：按照图4进行连接	
测试步骤： 测试步骤 1： 测试 DUT 在链路维持阶段收到无效的维持连接请求时是否响应正确。 1. tTester发送激活命令INQUIRY(t) <sub>00</sub> ； 2. tTester在8ms内接收激活响应ATI <sub>00</sub> ； 3. tTester发送的连接命令CONNECT_REQ(t) <sub>00</sub> ； 4. tTester在8ms内接收连接响应CONNECT_RSP <sub>00</sub> ； 5. tTester在MC通道持续发送CHECK1_REQ(t) <sub>00</sub> 或者CHECK2_REQ(t) <sub>00</sub> ； 6. tTester发送无效的LINKCTL_REQ(e) <sub>00</sub> ； 7. tTester在8ms内接收DUT响应； 8. tTester发送APDATA_REQ(t1) <sub>00</sub> ； 9. tTester在8ms内接收DUT响应。	
测试流程 1	
响应方测试仪tTester	DUT
INQUIRY(t) <sub>00</sub>	ATI <sub>00</sub>
CONNECT_REQ(t) <sub>00</sub>	CONNECT_RSP <sub>00</sub>
CHECK1_REQ(t) <sub>00</sub> or CHECK2_REQ(t) <sub>00</sub>	
LINKCTL_REQ(e) <sub>00</sub>	Mute
APDATA_REQ(t1) <sub>00</sub>	Mute
测试步骤 2： 测试 DUT 在链路维持阶段段且磁通道没有消息的条件下，连续三次收到维持连接请求，对第三次维持连接请求的响应是否正确。 1. tTester发送激活命令INQUIRY(t) <sub>01</sub> ； 2. tTester在8ms内接收激活响应ATI <sub>01</sub> ； 3. tTester发送连接命令CONNECT_REQ(t) <sub>01</sub> ； 4. tTester在8ms内接收连接响应CONNECT_RSP <sub>01</sub> ； 5. tTester在44ms后发送LINKCTL_REQ(t) <sub>01</sub> ，之后在8ms内接收DUT响应LINKCTL_RSP <sub>01</sub> ；	



6. tTester在44ms后发送LINKCTL\_REQ(t)<sub>01</sub>，之后在8ms内接收DUT响应LINKCTL\_RSP<sub>01</sub>；
7. tTester在44ms后发送LINKCTL\_REQ(t)<sub>01</sub>，之后在8ms内接收DUT响应LINKCTL\_RSP<sub>01</sub>，检查LINKCTL\_RSP<sub>01</sub>的数据格式。

测试流程 2

响应方测试仪tTester		DUT
INQUIRY(t) <sub>01</sub>	————→	
	←————	ATI <sub>01</sub>
CONNECT_REQ(t) <sub>01</sub>	————→	
	←————	CONNECT_RSP <sub>01</sub>
LINKCTL_REQ(t) <sub>01</sub>	————→	
	←————	LINKCTL_RSP <sub>01</sub>
LINKCTL_REQ(t) <sub>01</sub>	————→	
	←————	LINKCTL_RSP <sub>01</sub>
INKCTL_REQ(t) <sub>01</sub>	————→	
	←————	LINKCTL_RSP <sub>01</sub>

测试步骤 3:

测试 DUT 在链路维持阶段且磁通道存在冲突检测请求 CHECK1\_REQ 的条件下，收到多次维持连接请求时响应是否正确，之后收到数据交换指令是否响应正确。

1. tTester发送激活命令INQUIRY(t)<sub>02</sub>；
2. tTester在8ms内接收激活响应ATI<sub>02</sub>；
3. tTester发送连接命令CONNECT\_REQ(t)<sub>02</sub>；
4. tTester在8ms内接收连接响应CONNECT\_RSP<sub>02</sub>；
5. tTester在MC通道持续发送CHECK1\_REQ(t)<sub>02</sub>；
6. tTester在44ms后发送LINKCTL\_REQ(t)<sub>02</sub>，之后在8ms内收DUT响应LINKCTL\_RSP(t)<sub>02</sub>；
7. tTester在44ms后发送LINKCTL\_REQ(t)<sub>02</sub>，之后在8ms内收DUT响应LINKCTL\_RSP(t)<sub>02</sub>；
8. tTester在44ms后发送LINKCTL\_REQ(t)<sub>02</sub>，之后在8ms内收DUT响应LINKCTL\_RSP(t)<sub>02</sub>，检查LINKCTL\_RSP<sub>01</sub>的数据格式；
9. tTester发送APDATA\_REQ(t)<sub>02</sub>；
10. tTester在500ms内接收DUT响应。

测试流程 3

响应方测试仪tTester		DUT
INQUIRY(t) <sub>02</sub>	————→	
	←————	ATI <sub>02</sub>
CONNECT_REQ(t) <sub>02</sub>	————→	
	←————	CONNECT_RSP <sub>02</sub>
CHECK1_REQ(t) <sub>02</sub>	————→	
LINKCTL_REQ(t) <sub>02</sub>	————→	
	←————	LINKCTL_RSP <sub>02</sub>

LINKCTL_REQ(t) <sub>02</sub>	————→	
	←————	LINKCTL_RSP <sub>02</sub>
INKCTL_REQ(t) <sub>02</sub>	————→	
	←————	LINKCTL_RSP <sub>02</sub>
APDATA_REQ(t1) <sub>02</sub>	————→	
	←————	APDATA_RSP <sub>02</sub>

测试步骤 4:

测试 DUT 在链路维持阶段且磁通道存在连接确认消息 CHECK2\_REQ 的条件下，收到多次维持连接请求时响应是否正确，之后收到数据交换指令是否响应正确。

1. tTester发送激活命令INQUIRY(t)<sub>03</sub>；
2. tTester在8ms内接收激活响应ATI<sub>03</sub>；
3. tTester发送连接命令CONNECT\_REQ(t)<sub>03</sub>；
4. tTester在8ms内接收连接响应CONNECT\_RSP<sub>03</sub>；
5. tTester在MC通道持续发送CHECK2\_REQ(t)<sub>03</sub>；
6. tTester在44ms后发送LINKCTL\_REQ(t)<sub>03</sub>，之后在8ms内接收DUT响应LINKCTL\_RSP<sub>03</sub>；
7. tTester在44ms后发送LINKCTL\_REQ(t)<sub>03</sub>，之后在8ms内接收DUT响应LINKCTL\_RSP<sub>03</sub>；
8. tTester在44ms后发送LINKCTL\_REQ(t)<sub>03</sub>，之后在8ms内接收DUT响应LINKCTL\_RSP<sub>03</sub>，检查LINKCTL\_RSP<sub>03</sub>的数据格式；
9. tTester发送APDATA\_REQ(t1)<sub>03</sub>；
10. tTester在500ms内接收DUT响应。

测试流程 4

响应方测试仪tTester		DUT
INQUIRY(t) <sub>03</sub>	————→	
	←————	ATI <sub>03</sub>
CONNECT_REQ(t) <sub>03</sub>	————→	
	←————	CONNECT_RSP <sub>03</sub>
CHECK2_REQ(t) <sub>03</sub>	————→	
LINKCTL_REQ(t) <sub>03</sub>	————→	
	←————	LINKCTL_RSP <sub>03</sub>
LINKCTL_REQ(t) <sub>03</sub>	————→	
	←————	LINKCTL_RSP <sub>03</sub>
LINKCTL_REQ(t) <sub>03</sub>	————→	
	←————	LINKCTL_RSP <sub>03</sub>
APDATA_REQ(t1) <sub>03</sub>	————→	
	←————	APDATA_RSP(t1) <sub>03</sub>

测试步骤 5:

测试 DUT 在链路维持阶段接收到“CDC”错误的冲突检测命令 CHECK1\_REQ(e)时是否正确地更新消息的状态“Status”

- 1. tTester发送激活命令INQUIRY(t)<sub>04</sub> ;
- 2. tTester在8ms内接收激活响应ATI<sub>04</sub> ;
- 3. tTester发送连接命令CONNECT\_REQ(t)<sub>04</sub> ;
- 4. tTester在8ms内接收连接响应CONNECT\_RSP<sub>04</sub> ;
- 5. tTester在MC通道持续发送CHECK1\_REQ(e)<sub>04</sub> ;
- 6. tTester发送LINKCTL\_REQ(t)<sub>04</sub> ;
- 7. tTester在8ms内接收LINKCTL\_RSP(t)<sub>04</sub>, 检查其数据格式。

测试流程 5

响应方测试仪tTester		DUT
INQUIRY(t) <sub>04</sub>	————→	
	←————	ATI <sub>04</sub>
CONNECT_REQ(t) <sub>04</sub>	————→	
	←————	CONNECT_RSP <sub>04</sub>
CHECK1_REQ(e) <sub>04</sub>	————→	
LINKCTL_REQ(t) <sub>04</sub>	————→	
	←————	LINKCTL_RSP <sub>04</sub>

测试步骤 6:

测试 DUT 在链路维持阶段阶段接收到“CDC”错误的冲突检测命令 CHECK2\_REQ(e)时是否正确地更新消息的状态“Status”。

- 1. tTester发送激活命令INQUIRY(t)<sub>05</sub> ;
- 2. tTester在8ms内接收激活响应ATI<sub>05</sub> ;
- 3. tTester发送连接命令CONNECT\_REQ(t)<sub>05</sub> ;
- 4. tTester在8ms内接收连接响应CONNECT\_RSP<sub>05</sub> ;
- 5. tTester在MC通道持续发送CHECK2\_REQ(e)<sub>05</sub> ;
- 6. tTester发送LINKCTL\_REQ(t)<sub>05</sub> ;
- 7. tTester在8ms内接收LINKCTL\_RSP(t)<sub>05</sub>, 检查其数据格式。

测试流程 6

响应方测试仪tTester		DUT
INQUIRY(t) <sub>03</sub>	————→	
	←————	ATI <sub>03</sub>
CONNECT_REQ(t) <sub>03</sub>	————→	
	←————	CONNECT_RSP <sub>03</sub>
CHECK2_REQ(e) <sub>03</sub>	————→	
LINKCTL_REQ(t) <sub>03</sub>	————→	
	←————	LINKCTL_RSP <sub>03</sub>



- 预期结果:
- 1. 执行测试步骤1~6, 响应方DUT的响应与测试流程1~6一致。
  - 2. LINKCTL\_RSP<sub>02</sub>、LINKCTL\_RSP<sub>03</sub>数据格式符合下表要求。
  - 3. LINKCTL\_RSP<sub>01</sub>、LINKCTL\_RSP<sub>04</sub>和LINKCTL\_RSP<sub>05</sub>的‘Status’值为0x01, 其余内容与下表相同:

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0x00或者0xD0h~0xFFh
MsgCode(8bit)	24
MsgLen(16 bit)	2
RandData (1 byte)	随机
Reserved (1 byte)	0
Checksum (2 byte)	对消息头和消息体数据进行Checksum校验

6.8.5 关闭连接测试

测试编号：6.8.5																																				
测试项目：关闭连接测试																																				
测试目的： 验证响应方DUT在关闭连接阶段的数据格式、逻辑操作和响应时间是否符合要求																																				
预置条件： 按照图 4 进行连接																																				
测试步骤： 测试步骤 1： 测试 DUT 结束阶段收到错误的关闭连接请求时响应是否正确。 1. tTester发送激活命令INQUIRY(t) <sub>00</sub> ； 2. tTester在8ms内接收激活响应ATI <sub>00</sub> ； 3. tTester发送的连接命令CONNECT_REQ <sub>00</sub> ； 4. tTester在8ms内接收连接响应CONNECT_RSP <sub>00</sub> ； 5. tTester在MC通道持续发送CHECK1_REQ(t) <sub>00</sub> 或者CHECK2_REQ(t) <sub>00</sub> ； 6. tTester发送APDATA_REQ(t1) <sub>00</sub> ； 7. tTester在500ms内接收DUT响应APDATA_RSP <sub>00</sub> ； 8. tTester发送CLOSE_REQ(e) <sub>00</sub> ； 9. tTester在500ms内接收DUT响应； 10. tTester发送APDATA_REQ(t1) <sub>00</sub> ； 11. tTester在500ms内接收DUT响应。																																				
测试流程 1																																				
<table><tr><td>响应方测试仪tTester</td><td></td><td>DUT</td></tr><tr><td>INQUIRY(t)<sub>00</sub></td><td>————&gt;</td><td></td></tr><tr><td></td><td>&lt;————</td><td>ATI<sub>00</sub></td></tr><tr><td>CONNECT_REQ(t)<sub>00</sub></td><td>————&gt;</td><td></td></tr><tr><td></td><td>&lt;————</td><td>CONNECT_RSP<sub>00</sub></td></tr><tr><td>CHECK1_REQ(t)<sub>01</sub> or CHECK2_REQ(t)<sub>01</sub></td><td>————&gt;</td><td></td></tr><tr><td>APDATA_REQ(t1)<sub>00</sub></td><td>————&gt;</td><td></td></tr><tr><td></td><td>&lt;————</td><td>APDATA_RSP<sub>00</sub></td></tr><tr><td>CLOSE_REQ(e)<sub>00</sub></td><td>————&gt;</td><td></td></tr><tr><td></td><td>&lt;————</td><td>Mute</td></tr><tr><td>APDATA_REQ(t1)<sub>00</sub></td><td>————&gt;</td><td></td></tr><tr><td></td><td>&lt;————</td><td>Mute</td></tr></table>	响应方测试仪tTester		DUT	INQUIRY(t) <sub>00</sub>	————>			<————	ATI <sub>00</sub>	CONNECT_REQ(t) <sub>00</sub>	————>			<————	CONNECT_RSP <sub>00</sub>	CHECK1_REQ(t) <sub>01</sub> or CHECK2_REQ(t) <sub>01</sub>	————>		APDATA_REQ(t1) <sub>00</sub>	————>			<————	APDATA_RSP <sub>00</sub>	CLOSE_REQ(e) <sub>00</sub>	————>			<————	Mute	APDATA_REQ(t1) <sub>00</sub>	————>			<————	Mute
响应方测试仪tTester		DUT																																		
INQUIRY(t) <sub>00</sub>	————>																																			
	<————	ATI <sub>00</sub>																																		
CONNECT_REQ(t) <sub>00</sub>	————>																																			
	<————	CONNECT_RSP <sub>00</sub>																																		
CHECK1_REQ(t) <sub>01</sub> or CHECK2_REQ(t) <sub>01</sub>	————>																																			
APDATA_REQ(t1) <sub>00</sub>	————>																																			
	<————	APDATA_RSP <sub>00</sub>																																		
CLOSE_REQ(e) <sub>00</sub>	————>																																			
	<————	Mute																																		
APDATA_REQ(t1) <sub>00</sub>	————>																																			
	<————	Mute																																		
测试步骤 2： 测试 DUT 结束阶段收到需要回复的关闭连接请求时响应是否正确。 1. tTester发送激活命令INQUIRY(t) <sub>01</sub> ；																																				

- 2. tTester在8ms内接收激活响应ATI<sub>01</sub>；
- 3. tTester发送的连接命令CONNECT\_REQ<sub>01</sub>；
- 4. tTester在8ms内接收连接响应CONNECT\_RSP<sub>01</sub>；
- 5. tTester在MC通道持续发送CHECK1\_REQ(t)<sub>01</sub>或者CHECK2\_REQ(t)<sub>01</sub>；
- 6. tTester发送APDATA\_REQ(t1)<sub>01</sub>；
- 7. tTester在500ms内接收DUT响应APDATA\_RSP<sub>01</sub>；
- 8. tTester发送CLOSE\_REQ(t1)<sub>01</sub>；
- 9. tTester在500ms内接收并检查DUT响应CLOSE\_RSP<sub>01</sub>；
- 10. tTester发送APDATA\_REQ(t1)<sub>01</sub>；
- 11. tTester在500ms内接收DUT响应。

测试流程 2

响应方测试仪tTester		DUT
INQUIRY(t) <sub>01</sub>	————→	
	←————	ATI <sub>01</sub>
CONNECT_REQ(t) <sub>01</sub>	————→	
	←————	CONNECT_RSP <sub>01</sub>
CHECK1_REQ(t) <sub>01</sub> or CHECK2_REQ(t) <sub>01</sub>	————→	
APDATA_REQ(t1) <sub>01</sub>	————→	
	←————	APDATA_RSP <sub>01</sub>
CLOSE_REQ(t1) <sub>01</sub>	————→	
	←————	CLOSE_RSP <sub>01</sub>
APDATA_REQ(t1) <sub>01</sub>	————→	
	←————	Mute

测试步骤 3:

测试 DUT 结束阶段收到不需要回复的关闭连接请求时响应是否正确。

- 1. tTester发送激活命令INQUIRY(t)<sub>02</sub>；
- 2. tTester在8ms内接收激活响应ATI<sub>02</sub>；
- 3. tTester发送的连接命令CONNECT\_REQ<sub>02</sub>；
- 4. tTester在8ms内接收连接响应CONNECT\_RSP<sub>02</sub>；
- 5. tTester在MC通道持续发送CHECK1\_REQ(t)<sub>02</sub>或者CHECK2\_REQ(t)<sub>02</sub>；
- 6. tTester发送APDATA\_REQ(t1)<sub>02</sub>；
- 7. tTester在500ms内接收DUT响应APDATA\_RSP<sub>02</sub>；
- 8. tTester发送CLOSE\_REQ(t2)<sub>02</sub>；
- 9. tTester在500ms内接收DUT响应；
- 10. tTester发送APDATA\_REQ(t1)<sub>02</sub>；
- 11. tTester在500ms内接收DUT响应。



测试流程 3		
响应方测试仪tTester		DUT
INQUIRY(t) <sub>02</sub>	————→	
	←————	ATI <sub>02</sub>
CONNECT_REQ(t) <sub>02</sub>	————→	
	←————	CONNECT_RSP <sub>02</sub>
CHECK1_REQ(t) <sub>02</sub> or CHECK2_REQ(t) <sub>02</sub>	————→	
APDATA_REQ(t1) <sub>02</sub>	————→	
	←————	APDATA_RSP <sub>02</sub>
CLOSE_REQ(t2) <sub>00</sub>	————→	
	←————	Mute
APDATA_REQ(t1) <sub>02</sub>	————→	
	←————	Mute

预期结果：

1. 执行测试步骤1~3，响应方DUT的响应与测试流程一致。

2. CLOSE\_RSP<sub>01</sub>数据格式符合下表要求：

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0x00或者0xD0h~0xFFh
MsgCode(8bit)	27
MsgLen(16 bit)	4
CloseResult (1 byte)	0
Reserved(3 byte)	0
Checksum(2 byte)	对消息头和消息体数据进行Checksum校验。

6.8.6 冲突机制测试

测试编号：6.8.6	
测试项目：冲突机制测试	
测试目的： 验证响应方DUT冲突检测机制的数据格式、逻辑操作和响应时间是否符合要求	
预置条件：按照图4进行连接	
测试步骤： 测试步骤 1： 验证 DUT 在未连接状态下收到 CHECK1_REQ 是否在冲突响应时间窗（4ms）内随机地发送冲突响应消息。 1. tTester持续发送CHECK1_REQ(t) <sub>00</sub> ； 2. tTester每4ms内接收CHECK_RSP <sub>00</sub> ，若收到一次CHECK_RSP <sub>00</sub> 则测试Pass，否则测试Fail； 3. 重复200次执行步骤2。	
测试流程	
响应方测试仪tTester	DUT
CHECK1_REQ(t) <sub>00</sub>	CHECK_RSP <sub>00</sub>
预期结果： 1. 执行以上测试步骤，响应方DUT的响应与测试流程一致。 2. 测试过程中不出现Fail情况。 3. CHECK_RSP <sub>00</sub> 的数据格式符合下表要求。	
字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0x01
MsgCode(8bit)	24
MsgLen(16 bit)	8
RandData(1 byte)	随机数
Reserved(7 byte)	0
Checksum(2 byte)	对消息头和消息体数据进行Checksum校验

## 6.8.7 通道绑定测试

测试编号：6.8.7	
测试项目：通道绑定测试	
测试目的： 验证响应方DUT磁通道和射频通道之间的绑定机制是否有效，即验证DUT在维持连接、数据交易、激活和连接等不同阶段未接收到磁通道消息的情况下，DUT在射频通道上的响应是否符合要求	
预置条件：按照图4进行连接	
测试步骤： 测试步骤 1： 测试 DUT 维持连接阶段未接收到磁通道消息时，是否正确更新响应消息的状态“Status”。 1. tTester发送激活命令INQUIRY(t) <sub>00</sub> ； 2. tTester在8ms内接收DUT的响应ATI <sub>00</sub> ； 3. tTester发送连接命令CONNECT_REQ(t) <sub>00</sub> ； 4. tTester在8ms内接收DUT的响应CONNECT_RSP <sub>00</sub> ； 5. tTester关闭磁场； 6. tTester发送LINKCTL_REQ <sub>00</sub> ； 7. tTester在8ms内接收LINKCTL_RSP <sub>00</sub> ，检查其Status字段。	
测试流程 1	
响应方测试仪tTester	DUT
INQUIRY(t) <sub>00</sub>	ATI <sub>00</sub>
CONNECT_REQ(t) <sub>00</sub>	CONNECT_RSP <sub>00</sub>
关闭磁场	
LINKCTL_REQ(t) <sub>00</sub>	LINKCTL_RSP(t) <sub>00</sub>
测试步骤 2： 测试 DUT 数据交易阶段未接收到磁通道消息时，是否正确更新响应消息的状态“Status”。 1. tTester发送激活命令INQUIRY(t) <sub>01</sub> ； 2. tTester在8ms内接收DUT的响应ATI <sub>01</sub> ； 3. tTester发送连接命令CONNECT_REQ(t) <sub>01</sub> ； 4. tTester在8ms内接收DUT的响应CONNECT_RSP <sub>01</sub> ； 5. tTester关闭磁场； 6. tTester发送APDATA_REQ <sub>01</sub> ； 7. tTester在8ms内接收APDATA_RSP <sub>01</sub> ，检查其Status字段。	

测试流程 2	
响应方测试仪tTester	DUT
INQUIRY(t) <sub>01</sub>	ATI <sub>01</sub>
CONNECT_REQ(t) <sub>01</sub>	CONNECT_RSP <sub>01</sub>
关闭磁场	
APDATA_REQ(t) <sub>01</sub>	APDATA_RSP(t) <sub>01</sub>
测试步骤 3:	
测试 DUT 激活阶段未接收到磁通道消息时, 响应是否符合要求。	
1. tTester发送激活命令INQUIRY(t) <sub>02</sub> ;	
2. tTester关闭磁场;	
3. tTester在8ms内接收DUT的响应, 若收到DUT的射频消息则测试Fail, 否则测试Pass。	
测试流程 3	
响应方测试仪tTester	DUT
INQUIRY(t) <sub>02</sub>	
关闭磁场	Mute
测试步骤 4:	
测试 DUT 连接阶段未接收到磁通道消息时, 响应是否符合要求。	
1. tTester发送激活命令INQUIRY(t) <sub>03</sub> ;	
2. tTester在8ms内接收DUT的响应ATI <sub>03</sub> ;	
3. tTester关闭磁场;	
4. tTester发送CONNECT_REQ(t) <sub>03</sub> ;	
5. tTester在8ms内接收DUT的响应, 若收到DUT的射频消息则测试Fail, 否则测试Pass。	
测试流程 4	
响应方测试仪tTester	DUT
INQUIRY(t) <sub>03</sub>	ATI <sub>03</sub>
关闭磁场	
CONNECT_REQ(t) <sub>03</sub>	Mute
预期结果:	
1. 执行上述测试步骤1~4, 不出现测试Fail的情况。	
2. LINKCTL_RSP <sub>00</sub> 的Status字段为0x82或0x02或0x01。	
3. APDATA_RSP <sub>01</sub> 的Status字段为0x82或0x02	



6.8.8 超时时间测试

测试编号：6.8.8	
测试项目：超时时间测试	
测试目的： 验证响应方DUT在连接阶段、链路维持阶段、数据交易阶段的超时时间是否符合要求	
预置条件：按照图4进行连接	
测试步骤： 测试步骤 1： 测试 DUT 在连接阶段的超时时间是否与 YD/T XXX.10 一致。 1. tTester发送激活命令INQUIRY(t) <sub>00</sub> ； 2. tTester在8ms内接收DUT的响应ATI <sub>00</sub> ； 3. tTester在收到ATI <sub>00</sub> 时立即启动计时，第7ms发送CONNECT_REQ(t) <sub>00</sub> ； 4. tTester在8ms内接收DUT的响应，若接收到CONNECT_RSP则测试Pass，否则测试Fail。	
测试流程 1	
响应方测试仪tTester	DUT
INQUIRY(t) <sub>00</sub>	ATI <sub>00</sub>
CONNECT_REQ(t) <sub>00</sub>	CONNECT_RSP <sub>00</sub>
测试步骤 2： 测试 DUT 在连接阶段的超时时间是否与 YD/T XXX.10 一致。 1. tTester发送激活命令INQUIRY(t) <sub>01</sub> ； 2. tTester在8ms内接收DUT的响应ATI <sub>01</sub> ； 3. tTester在收到ATI <sub>00</sub> 时立即启动计时，第10ms发送CONNECT_REQ <sub>01</sub> ； 4. tTester在8ms内接收DUT的响应，若DUT无响应则测试Pass，否则测试Fail。	
测试流程 2	
响应方测试仪tTester	DUT
INQUIRY(t) <sub>01</sub>	ATI <sub>01</sub>
CONNECT_REQ(t) <sub>01</sub>	Mute
测试步骤 3： 测试 DUT 在链路维持阶段的超时时间是否与 YD/T XXX.10 一致。 1. tTester发送激活命令INQUIRY(t) <sub>02</sub> ； 2. tTester在8ms内接收DUT的响应ATI <sub>02</sub> ；	

3. tTester在收到DUT的响应ATI<sub>02</sub>后发送CONNECT\_REQ(t)<sub>02</sub> ;
4. tTester在8ms内接收DUT的CONNECT\_RSP<sub>02</sub> ;
5. tTester的MC通道持续发送CHECK1\_REQ(t)<sub>02</sub>或者CHECK2\_REQ(t)<sub>02</sub> ;
6. tTester在收到CONNECT\_RSP<sub>02</sub>时立即启动计时, 第80ms发送LINKCTL\_REQ(t)<sub>02</sub> ;
7. tTester在8ms内接收DUT的响应, 若接收到LINKCTL\_RSP<sub>02</sub>则测试Pass, 否则测试Fail。

测试流程 3

响应方测试仪tTester	DUT
INQUIRY(t) <sub>02</sub>	ATI <sub>02</sub>
CONNECT_REQ(t) <sub>02</sub>	CONNECT_RSP <sub>02</sub>
CHECK1_REQ(t) <sub>02</sub> or CHECK2_REQ(t) <sub>02</sub>	
LINKCTL_REQ(t) <sub>02</sub>	LINKCTL_RSP <sub>02</sub>

测试步骤 4:

测试 DUT 在链路维持阶段的超时时间是否与 YD/T XXX.10 一致。

1. tTester发送激活命令INQUIRY(t)<sub>03</sub> ;
2. tTester在8ms内接收DUT的响应ATI<sub>03</sub> ;
3. tTester在收到DUT的响应ATI<sub>03</sub>后发送CONNECT\_REQ(t)<sub>03</sub> ;
4. tTester在8ms内接收DUT的CONNECT\_RSP<sub>03</sub> ;
5. tTester的MC通道发送CHECK1\_REQ(t)<sub>03</sub>或者CHECK2\_REQ(t)<sub>03</sub> ;
6. tTester在收到CONNECT\_RSP<sub>03</sub>时立即启动计时, 第120ms发送LINKCTL\_REQ(t)<sub>03</sub> ;
7. tTester在8ms内接收DUT的响应, 若DUT无响应则测试Pass, 否则测试Fail。

测试流程 4

响应方测试仪tTester	DUT
INQUIRY(t) <sub>03</sub>	ATI <sub>03</sub>
CONNECT_REQ(t) <sub>03</sub>	CONNECT_RSP <sub>03</sub>
CHECK1_REQ(t) <sub>03</sub> or CHECK2_REQ(t) <sub>03</sub>	
LINKCTL_REQ(t) <sub>03</sub>	Mute

测试步骤 5:

测试 DUT 在数据交易阶段的超时时间是否与 YD/T XXX.10 一致。

1. tTester发送激活命令INQUIRY(t)<sub>04</sub> ;

- 2. tTester在8ms内接收DUT的响应ATI<sub>04</sub>；
- 3. tTester在收到DUT的响应ATI<sub>00</sub>后发送CONNECT\_REQ(t)<sub>04</sub>；
- 4. tTester在8ms内接收DUT的CONNECT\_RSP<sub>04</sub>；
- 5. tTester的MC通道发送CHECK1\_REQ(t)<sub>04</sub>或者CHECK2\_REQ(t)<sub>04</sub>；
- 6. tTester在收到CONNECT\_RSP<sub>04</sub>时立即启动计时，第80ms开始发送APDATA\_REQ(t)<sub>04</sub>；
- 7. tTester在500ms内接收DUT的响应，若接收到APDATA\_RSP则测试Pass，否则测试Fail。

测试流程 5

响应方测试仪tTester	DUT
INQUIRY(t) <sub>04</sub>	ATI <sub>04</sub>
CONNECT_REQ(t) <sub>04</sub>	CONNECT_RSP <sub>04</sub>
CHECK1_REQ(t) <sub>04</sub> or CHECK2_REQ(t) <sub>04</sub>	
APDATA_REQ(t) <sub>04</sub>	APDATA_RSP <sub>04</sub>

测试步骤 6:

测试 DUT 在数据交易阶段的超时时间是否与 YD/T XXX.10 一致。

- 1. tTester发送激活命令INQUIRY(t)<sub>05</sub>；
- 2. tTester在8ms内接收DUT的响应ATI<sub>05</sub>；
- 3. tTester在收到DUT的响应ATI<sub>05</sub>后发送CONNECT\_REQ(t)<sub>05</sub>；
- 4. tTester在8ms内接收DUT的CONNECT\_RSP<sub>05</sub>；
- 5. tTester的MC通道发送CHECK1\_REQ(t)<sub>05</sub>或者CHECK2\_REQ(t)<sub>05</sub>；
- 6. tTester在收到CONNECT\_RSP<sub>05</sub>时立即启动计时，第120ms发送APDATA\_REQ(t)<sub>05</sub>；
- 7. tTester在500ms内接收DUT的响应，若DUT无响应则测试Pass，否则测试Fail。

测试流程 6

响应方测试仪tTester	DUT
INQUIRY(t) <sub>05</sub>	ATI <sub>05</sub>
CONNECT_REQ(t) <sub>05</sub>	CONNECT_RSP <sub>05</sub>
CHECK1_REQ(t) <sub>05</sub> or CHECK2_REQ(t) <sub>05</sub>	
APDATA_REQ(t) <sub>05</sub>	Mute

预期结果:

执行测试步骤 1~6，不出现测试 Fail 情况



6.8.9 加密算法验证

测试编号: 6.8.9
测试项目: 加密算法验证
测试目的: 验证响应方 DUT 支持的加密算法是否符合要求
预置条件: 按照图4进行连接

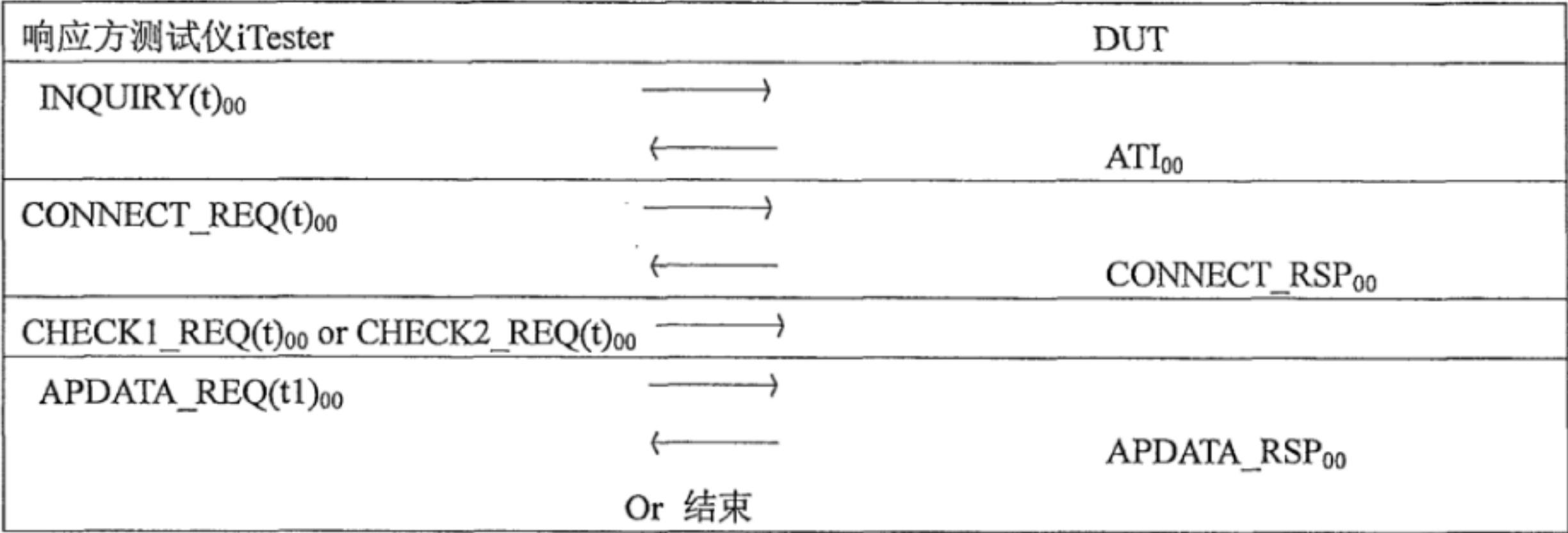
测试步骤:

测试步骤 1:

测试 DUT 的 3DES-CBC 算法。

1. tTester发送激活命令INQUIRY(t)<sub>00</sub> ;
2. tTester在8ms内接收DUT的响应ATI<sub>00</sub> ;
3. tTester在发送CONNECT\_REQ(t)<sub>00</sub>, ‘EncAlg’ 字段的值为 ‘0x0200’ ;
4. tTester在8ms内接收DUT的CONNECT\_RSP<sub>00</sub> ;
5. tTester在MC通道持续发送CHECK1\_REQ(t)<sub>00</sub>或者CHECK2\_REQ(t)<sub>00</sub> ;
6. tTester分析CONNECT\_RSP<sub>00</sub>, 若 ‘EncAlg’ 字段的值为 ‘0x0000’ 则说明DUT不支持该算法测试结束, 若为 ‘0x0200’ 则进入步骤9, 否则若为其他值则测试Fail ;
7. tTester发送APDATA\_REQ(t1)<sub>00</sub> ;
8. tTester在500ms内接收APDATA\_RSP<sub>00</sub>, 核对其数据的加密算法是否正确。

测试流程 1

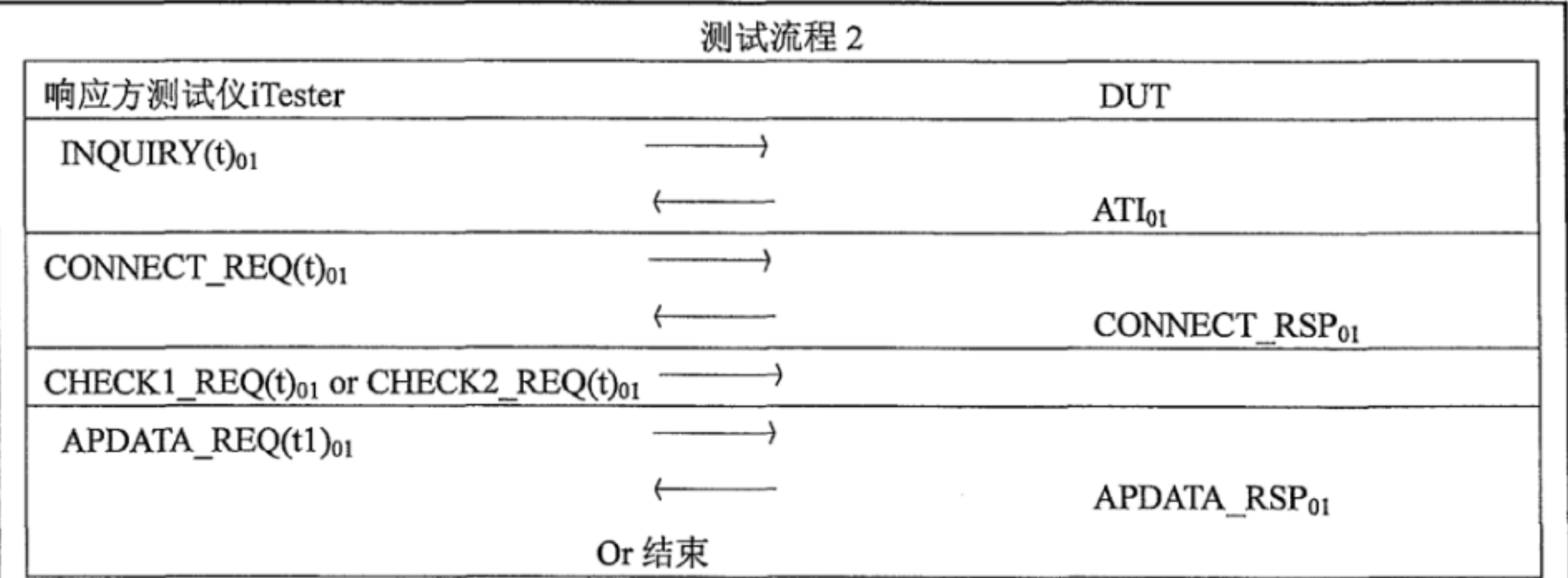


测试步骤 2:

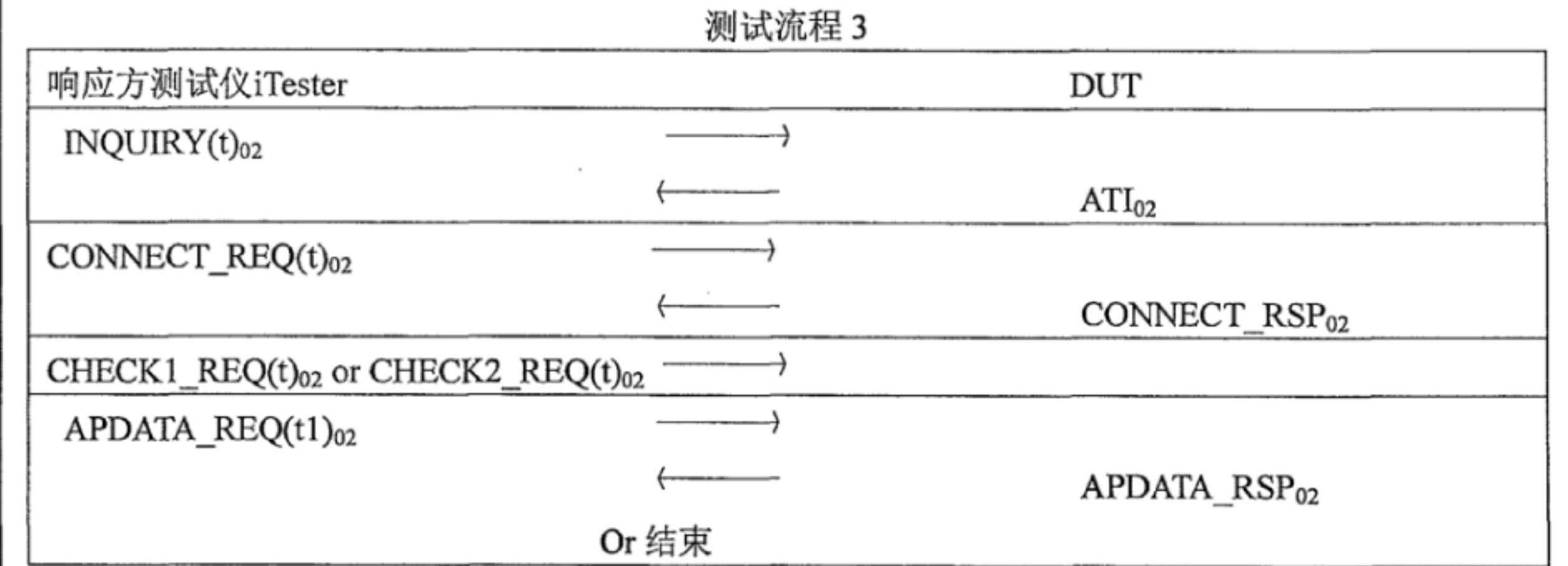
测试 DUT 的 AES-ECB 算法。

1. tTester发送激活命令INQUIRY(t)<sub>01</sub> ;
2. tTester在8ms内接收DUT的响应ATI<sub>01</sub> ;
3. tTester在收到DUT的响应ATI<sub>01</sub>后发送CONNECT\_REQ(t)<sub>01</sub>, ‘EncAlg’ 字段的值为 ‘0x0400’ ;
4. tTester在8ms内接收到DUT的CONNECT\_RSP<sub>01</sub> ;
5. tTester在MC通道持续发送CHECK1\_REQ(t)<sub>01</sub>或者CHECK2\_REQ(t)<sub>01</sub> ;
6. tTester分析CONNECT\_RSP<sub>01</sub>, 若 ‘EncAlg’ 字段的值为 ‘0x0000’ 则说明DUT不支持该算法测试结束, 若为 ‘0x0400’ 则进入步骤8, 否则若为其他值则测试Fail;
7. tTester发送APDATA\_REQ(t1)<sub>01</sub> ;
8. tTester在500ms内接收APDATA\_RSP<sub>01</sub>, 核对其数据的加密算法是否正确。





- 测试步骤 3:
- 测试 DUT 的 AES-CBC 算法。
1. tTester发送激活命令INQUIRY(t)<sub>02</sub>；
  2. tTester在8ms内接收DUT的响应ATI<sub>02</sub>；
  3. tTester在收到DUT的响应ATI<sub>02</sub>后发送CONNECT\_REQ(t)<sub>02</sub>， ‘EncAlg’ 字段的值为 ‘0x0800’；
  4. tTester在8ms内接收到DUT的CONNECT\_RSP<sub>02</sub>；
  5. tTester在MC通道持续发送CHECK1\_REQ(t)<sub>02</sub>或者CHECK2\_REQ(t)<sub>02</sub>；
  6. tTester分析CONNECT\_RSP<sub>02</sub>，若 ‘EncAlg’ 字段的值为 ‘0x0000’ 则说明DUT不支持该算法测试结束，若为 ‘0x0800’ 则进入步骤8，否则若为其他值则测试Fail；
  7. tTester发送APDATA\_REQ(t1)<sub>02</sub>；
  8. tTester在500ms内接收APDATA\_RSP<sub>02</sub>，核对其数据的加密算法是否正确。



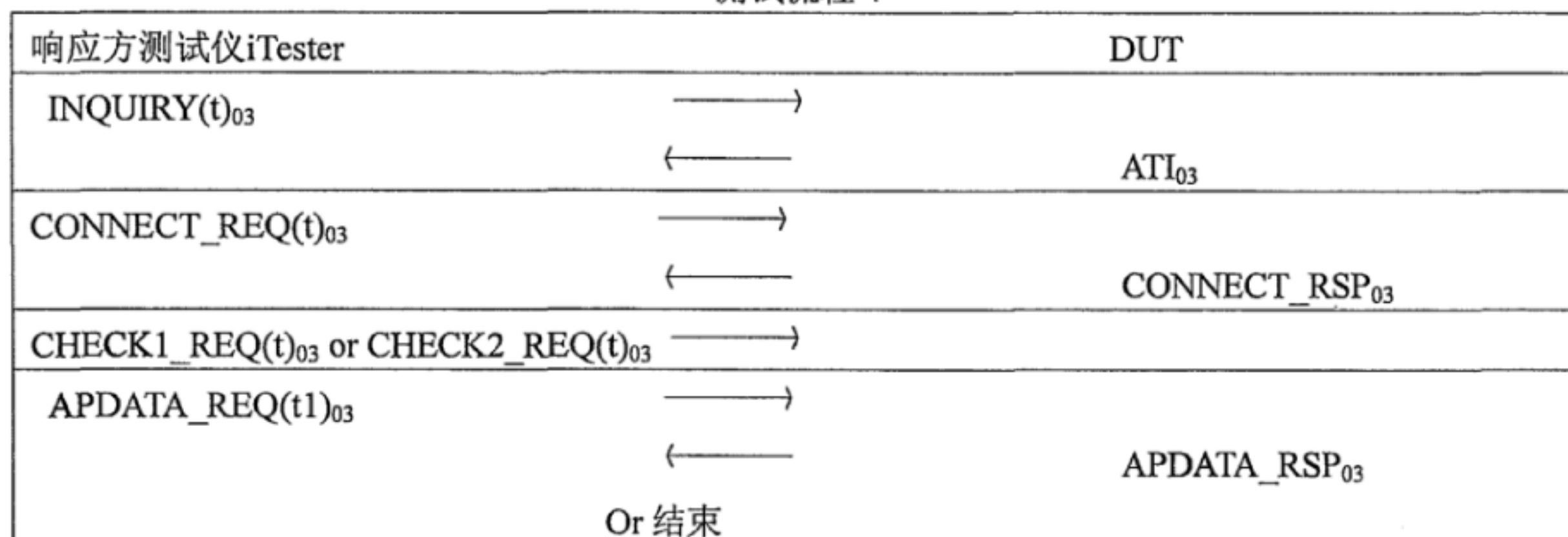
- 测试步骤 4:
- 测试 DUT 的 SM4-ECB 算法。
1. tTester发送激活命令INQUIRY(t)<sub>03</sub>；
  2. tTester在8ms内接收DUT的响应ATI<sub>03</sub>；
  3. tTester在收到DUT的响应ATI<sub>03</sub>后发送CONNECT\_REQ(t)<sub>03</sub>， ‘EncAlg’ 字段的值为 ‘0x0001’；
  4. tTester在8ms内接收到DUT的CONNECT\_RSP<sub>03</sub>；
  5. tTester在MC通道持续发送CHECK1\_REQ(t)<sub>03</sub>或者CHECK2\_REQ(t)<sub>03</sub>；
  6. tTester分析CONNECT\_RSP<sub>03</sub>，若 ‘EncAlg’ 字段的值为 ‘0x0000’ 则说明DUT不支持该算法测试结

束, 若为 '0x0001' 则进入步骤8, 否则若为其他值则测试Fail;

7. tTester发送APDATA\_REQ(t1)<sub>03</sub>;

8. tTester在500ms内接收APDATA\_RSP<sub>03</sub>, 核对其数据的加密算法是否正确。

#### 测试流程 4



测试步骤 5:

测试 DUT 的 SM4-CBC 算法。

1. tTester发送激活命令INQUIRY(t)<sub>04</sub>;

2. tTester在8ms内接收DUT的响应ATI<sub>04</sub>;

3. tTester在收到DUT的响应ATI<sub>04</sub>后发送CONNECT\_REQ(t)<sub>04</sub>, 'EncAlg' 字段的值为 '0x0002';

4. tTester在8ms内接收到DUT的CONNECT\_RSP<sub>04</sub>;

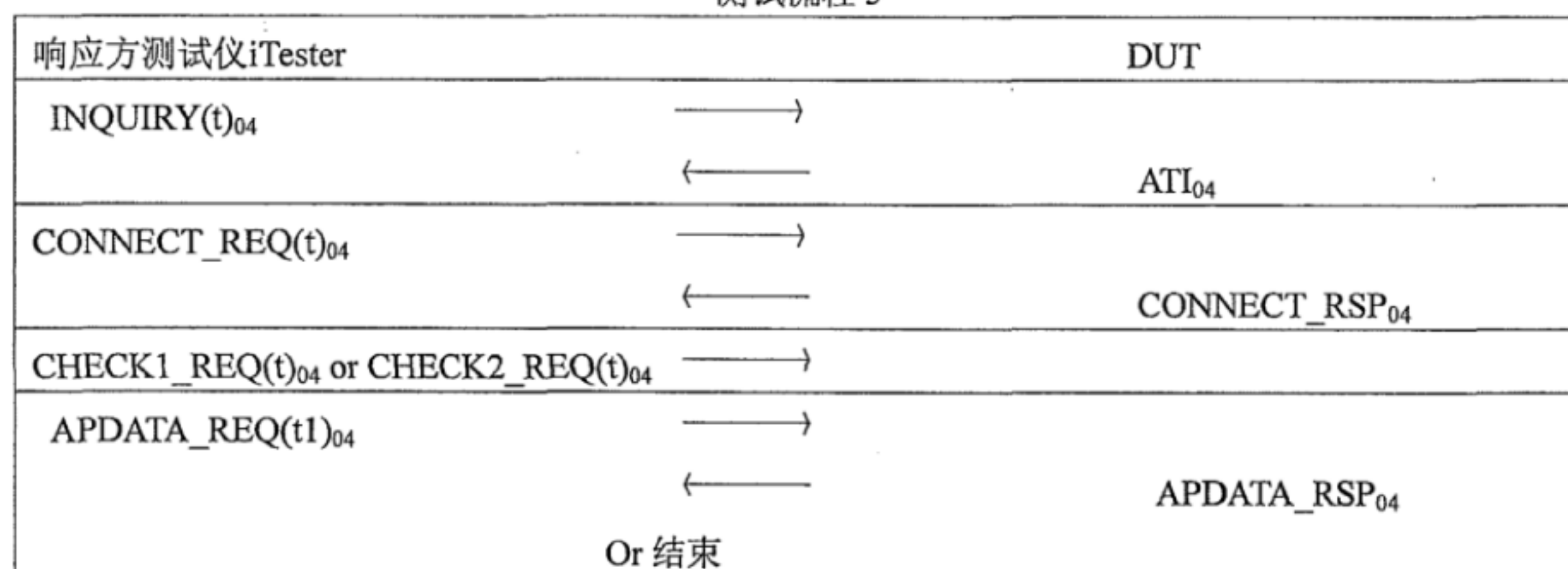
5. tTester在MC通道持续发送CHECK1\_REQ(t)<sub>04</sub>或者CHECK2\_REQ(t)<sub>04</sub>;

6. tTester分析CONNECT\_RSP<sub>04</sub>, 若 'EncAlg' 字段的值为 '0x0000' 则说明DUT不支持该算法测试结束, 若为 '0x0002' 则进入步骤8, 否则若为其他值则测试Fail;

7. tTester发送APDATA\_REQ(t1)<sub>04</sub>;

8. tTester在500ms内接收APDATA\_RSP<sub>04</sub>, 检查验证其数据的加密算法。

#### 测试流程 5



预期结果:

1. 执行上述测试步骤1~5, DUT按照测试流程进行响应不出现测试Fail情况。

2. 执行上述测试步骤1~5, 所有数据的加密算法验证通过

测试编号: 6.8.10

测试项目：频点范围测试

测试目的:

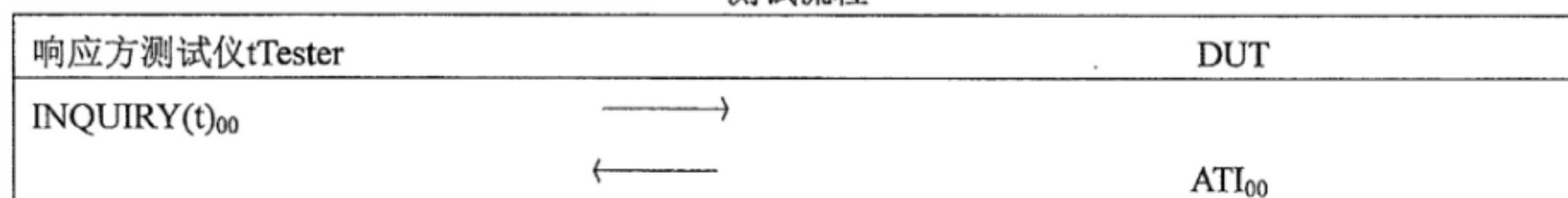
### 验证响应方DUT的频点范围是否符合要求

预置条件：按照图4进行连接

测试步骤:

1. tTester发送激活命令INQUIRY(t)<sub>00</sub>, 设置其‘IDm’使得ATI的频点为2401MHz;
2. tTester在8ms内接收激活响应ATI<sub>00</sub>, 若没有收到ATI<sub>00</sub>则测试Fail且结束;
3. 重复63次执行步骤2, 每重复一次设置其‘IDm’使得ATI的频点增加1MHz。

## 测试流程



预期结果:

执行上述测试步骤，不出现测试 Fail 情况

## 6.9 发起方协议测试项目

## 6.9.1 激活测试

测试编号: 6.9.1

测试项目: 激活测试

测试目的:

验证发起方DUT在激活阶段的数据格式、逻辑操作和响应时间是否符合要求

预置条件: 按照图5进行连接

测试步骤:

测试步骤 1:

测试 DUT 在激活阶段收到无效的 ATI 时处理是否正确。

- 1, DUT发送INQUIRY<sub>00</sub>, iTester接收并检查INQUIRY<sub>00</sub>的数据格式;
- 2, iTester在8ms内响应无效的ATI(e)<sub>00</sub>;
- 3, iTester在8ms内接收来自DUT的命令。

测试流程 1

DUT	发起方测试仪 iTester
INQUIRY <sub>00</sub>	→
	← ATI(e) <sub>00</sub>
Mute or INQUIRY	→

测试步骤 2:

测试 DUT 在激活阶段接收超时后处理是否正确。

1. DUT发送INQUIRY<sub>01</sub>;
2. iTester等待8ms, 不作响应;
3. iTester在8ms内接收来自DUT的命令。

测试流程 2

DUT	发起方测试仪 iTester
INQUIRY <sub>01</sub>	→
	← Mute
Mute or INQUIRY	→

测试步骤 3:

测试 DUT 在激活阶段接收到有效的 ATI 时处理是否正确。

1. DUT 发送 INQUIRY<sub>02</sub>;
2. iTester 在 8ms 内响应有效的 ATI(t)<sub>02</sub>;
3. iTester 在 8ms 内接收来自 DUT 发送连接请求 CONNECT\_REQ<sub>02</sub>。



测试流程 3	
DUT	发起方测试仪iTester
INQUIRY <sub>02</sub>	—————>
	<—————
	ATI(t) <sub>02</sub>
CONNECT_REQ <sub>02</sub>	—————>

预期结果：

1. 执行测试步骤1~3，DUT的响应与测试流程一致。

2. INQUIRY<sub>00</sub>的数据格式符合下表要求：

字段	值
MsgCode(4 bit)	0
MsgLen(4 bit)	15
Rfu(4bit)	0
AccessVersion(4bit)	0x03
IDm(14 byte)	随机数

6.9.2 连接测试

测试编号：6.9.2													
测试项目：连接测试													
测试目的： 验证发起方DUT在建立连接阶段的数据格式、逻辑操作和响应时间是否符合要求													
预置条件：按照图5进行连接													
测试步骤： 测试步骤 1： 测试 DUT 在连接阶段收到无效的连接响应时的处理是否正确。 1. DUT发送INQUIRY <sub>00</sub> ； 2. iTester在8ms内响应有效的ATI(t) <sub>00</sub> ； 3. iTester在8ms内接收DUT的CONNECT_REQ <sub>00</sub> ， iTester检查消息的格式； 4. iTester在8ms内响应无效的CONNECT_RSP(e) <sub>00</sub> ； 5. iTester在100ms内接收DUT的响应。													
测试流程 1													
<table><tr><th>DUT</th><th>发起方测试仪iTester</th></tr><tr><td>INQUIRY<sub>00</sub></td><td>————→</td></tr><tr><td></td><td>←———— ATI(t)<sub>00</sub></td></tr><tr><td>CONNECT_REQ<sub>00</sub></td><td>————→</td></tr><tr><td></td><td>←———— CONNECT_RSP(e)<sub>00</sub></td></tr><tr><td>INQUIRY or CLOSE_REQ or Mute</td><td>————→</td></tr></table>		DUT	发起方测试仪iTester	INQUIRY <sub>00</sub>	————→		←———— ATI(t) <sub>00</sub>	CONNECT_REQ <sub>00</sub>	————→		←———— CONNECT_RSP(e) <sub>00</sub>	INQUIRY or CLOSE_REQ or Mute	————→
DUT	发起方测试仪iTester												
INQUIRY <sub>00</sub>	————→												
	←———— ATI(t) <sub>00</sub>												
CONNECT_REQ <sub>00</sub>	————→												
	←———— CONNECT_RSP(e) <sub>00</sub>												
INQUIRY or CLOSE_REQ or Mute	————→												
测试步骤 2： 测试 DUT 在连接阶段接收超时后处理是否正确。 1. DUT发送INQUIRY <sub>01</sub> ； 2. iTester在8ms内响应有效的ATI(t) <sub>01</sub> ； 3. iTester在8ms内接收DUT的CONNECT_REQ <sub>01</sub> ； 4. iTester空闲等待8ms， 不作响应； 5. iTester接收100ms的DUT的响应。													
测试流程 2													
<table><tr><th>DUT</th><th>发起方测试仪iTester</th></tr><tr><td>INQUIRY<sub>01</sub></td><td>————→</td></tr><tr><td></td><td>←———— ATI(t)<sub>01</sub></td></tr><tr><td>CONNECT_REQ<sub>01</sub></td><td>————→</td></tr><tr><td></td><td>←———— Mute</td></tr><tr><td>INQUIRY or CLOSE_REQ or Mute</td><td>————→</td></tr></table>		DUT	发起方测试仪iTester	INQUIRY <sub>01</sub>	————→		←———— ATI(t) <sub>01</sub>	CONNECT_REQ <sub>01</sub>	————→		←———— Mute	INQUIRY or CLOSE_REQ or Mute	————→
DUT	发起方测试仪iTester												
INQUIRY <sub>01</sub>	————→												
	←———— ATI(t) <sub>01</sub>												
CONNECT_REQ <sub>01</sub>	————→												
	←———— Mute												
INQUIRY or CLOSE_REQ or Mute	————→												

测试步骤 3:

测试 DUT 在连接阶段收到正常的连接响应时处理是否正确。

- 1. DUT发送INQUIRY<sub>02</sub> ;
- 2. iTester在8ms内响应有效的ATI(t)<sub>02</sub> ;
- 3. iTester在8ms内接收DUT的CONNECT\_REQ<sub>02</sub> ;
- 4. iTester在8ms内响应有效的CONNECT\_RSP(t)<sub>02</sub> ;
- 5. iTester在100ms内接收DUT的消息。

测试流程 3

DUT	发起方测试仪iTester
INQUIRY <sub>02</sub>	ATI(t) <sub>02</sub>
CONNECT_REQ <sub>02</sub>	CONNECT_RSP(t) <sub>02</sub>
CHECK1_REQ or CHECK2_REQ	
LINKCTL_REQ <sub>02</sub> or APDATA_REQ <sub>02</sub>	

测试步骤 4:

测试 DUT 在连接阶段是否判断连接结果。

- 1. DUT发送INQUIRY<sub>03</sub> ;
- 2. iTester在8ms内响应有效的ATI(t)<sub>03</sub> ;
- 3. iTester在8ms内接收DUT的CONNECT\_REQ<sub>03</sub> ;
- 4. iTester空8ms内响应有效的CONNECT\_RSP<sub>03</sub>, ‘Result’ 值为0x01;
- 5. iTester在100ms内接收DUT的响应。

测试流程 4

DUT	发起方测试仪iTester
INQUIRY <sub>03</sub>	ATI(t) <sub>03</sub>
CONNECT_REQ <sub>03</sub>	CONNECT_RSP <sub>03</sub>
INQUIRY or CLOSE_REQ or Mute	

预期结果:

- 1. 执行上述测试步骤, DUT的响应与测试流程一致。
- 2. CONNECT\_REQ<sub>00</sub>命令数据格式符合下表要求:

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8

Status(8bit)	0x00或者0xD0h~0xFFh
MsgCode(8bit)	17
MsgLen(16 bit)	24
InitiatorType(1 byte)	'A'
InitiatorID(8 byte)	自定义
RootKeyIndex(1 byte)	自定义
SessionKey(1 byte)	b0为1
EncAlg(2 byte)	b0为1, b6~b15为0
MDInfo(5 byte)	自定义
Reserved(6 byte)	0
Checksum(2 byte)	对消息头和消息体数据进行Checksum校验

注意：在没有预置密钥的情况下 RootKeyIndex 的值应该为“0”



6.9.3 数据交换测试

测试编号：6.9.3	
测试项目：数据交换测试	
测试目的： 验证发起方DUT在数据交易阶段的数据格式、逻辑操作和响应时间是否符合要求	
预置条件：按照图5进行连接	
测试步骤： 测试步骤 1： 测试 DUT 在交易阶段收到错误的单包数据交易响应时处理是否正确。 1. DUT发送INQUIRY <sub>00</sub> ； 2. iTester在8ms内响应有效的ATI(t) <sub>00</sub> ； 3. iTester在8ms内接收DUT的CONNECT_REQ <sub>00</sub> ； 4. iTester在8ms内响应CONNECT_RSP(t) <sub>00</sub> ； 5. DUT在MC通道持续发送CHECK1_REQ(t) <sub>00</sub> 或者CHECK2_REQ(t) <sub>00</sub> ； 6. DUT发送APDATA_REQ(t1) <sub>00</sub> ，iTester检查其数据格式； 7. iTester在500ms内响应APDATA_RSP(e) <sub>00</sub> ； 8. iTester在100ms内接收DUT的消息。	
测试流程 1	
DUT	发起方测试仪iTester
INQUIRY <sub>00</sub>	ATI(t) <sub>00</sub>
CONNECT_REQ <sub>00</sub>	CONNECT_RSP(t) <sub>00</sub>
CHECK1_REQ(t) <sub>00</sub> or CHECK2_REQ(t) <sub>00</sub>	
APDATA_REQ(t1) <sub>00</sub>	APDATA_RSP(e) <sub>00</sub>
Mute or INQUIRY or CLOSE_REQ	Mute
测试步骤 2： 测试 DUT 在交易阶段收到正确的单包数据交易响应后处理是否正确。 1. DUT发送INQUIRY <sub>01</sub> ； 2. iTester在8ms内响应有效的ATI(t) <sub>01</sub> ； 3. iTester在8ms内接收DUT的CONNECT_REQ <sub>01</sub> ； 4. iTester在8ms内响应CONNECT_RSP(t) <sub>01</sub> ； 5. DUT在MC通道持续发送CHECK1_REQ(t) <sub>01</sub> 或者CHECK2_REQ(t) <sub>01</sub> ； 6. DUT发送APDATA_REQ(t1) <sub>01</sub> ； 7. iTester在500ms内响应APDATA_RSP(t) <sub>01</sub> ； 8. Tester在100ms内接收DUT的消息。	

## 测试流程 2

DUT	发起方测试仪 iTester
INQUIRY <sub>01</sub> —————>	
	<———— ATI(t) <sub>01</sub>
CONNECT_REQ <sub>01</sub> —————>	
	<———— CONNECT_RSP(t) <sub>01</sub>
CHECK1_REQ(t) <sub>01</sub> or CHECK2_REQ(t) <sub>01</sub> —————>	
APDATA_REQ(t) <sub>01</sub> —————>	
	<———— APDATA_RSP(t) <sub>01</sub>
LINKCTL_REQ <sub>01</sub> or APDATA_REQ <sub>01</sub> or CLOSE_REQ <sub>01</sub> —————>	

## 测试步骤 3:

测试 DUT 在交易阶段收到错误的长包数据交易响应后处理是否正确。

1. DUT 发送 INQUIRY<sub>02</sub> ;
2. iTester 在 8ms 内响应有效的 ATI(t)<sub>02</sub> ;
3. iTester 在 8ms 内接收 DUT 的 CONNECT\_REQ<sub>02</sub> ;
4. iTester 在 8ms 内响应 CONNECT\_RSP(t)<sub>02</sub> ;
5. DUT 在 MC 通道持续发送 CHECK1\_REQ(t)<sub>02</sub> 或者 CHECK2\_REQ(t)<sub>02</sub> ;
6. DUT 发送 APDATA\_REQ(t)<sub>02</sub>, iTester 检查其数据格式;
7. iTester 在 500ms 内响应 APDATA\_RSP(e)<sub>02</sub> ;
8. iTester 在 100ms 内接收 DUT 的消息。

## 测试流程 3

DUT	发起方测试仪 iTester
INQUIRY <sub>02</sub> —————>	
	<———— ATI(t) <sub>02</sub>
CONNECT_REQ <sub>02</sub> —————>	
	<———— CONNECT_RSP(t) <sub>02</sub>
CHECK1_REQ(t) <sub>02</sub> or CHECK2_REQ(t) <sub>02</sub> —————>	
APDATA_REQ(t) <sub>02</sub> —————>	
	<———— APDATA_RSP(e) <sub>02</sub>
Mute or INQUIRY or CLOSE_REQ —————>	

## 测试步骤 4:

测试 DUT 在交易阶段收到正确的长包数据交易响应后处理是否正确。

1. DUT 发送 INQUIRY<sub>03</sub> ;
2. iTester 在 8ms 内响应有效的 ATI(t)<sub>03</sub> ;
3. iTester 在 8ms 内接收 DUT 的 CONNECT\_REQ<sub>03</sub> ;
4. iTester 在 8ms 内响应 CONNECT\_RSP(t)<sub>03</sub> ;
5. DUT 在 MC 通道持续发送 CHECK1\_REQ(t)<sub>03</sub> 或者 CHECK2\_REQ(t)<sub>03</sub> ;
6. DUT 发送 APDATA\_REQ(t)<sub>03</sub> ;
7. iTester 在 500ms 内响应 APDATA\_RSP(t)<sub>03</sub> ;

8. iTester在100ms内接收DUT的消息。

测试流程 4

DUT	发起方测试仪iTester
INQUIRY <sub>03</sub>	ATI(t) <sub>03</sub>
CONNECT_REQ <sub>03</sub>	CONNECT_RSP(t) <sub>03</sub>
CHECK1_REQ(t) <sub>03</sub> or CHECK2_REQ(t) <sub>03</sub>	
APDATA_REQ(t2) <sub>03</sub>	APDATA_RSP(t) <sub>03</sub>
LINKCTL_REQ <sub>03</sub> or APDATA_REQ <sub>03</sub> or CLOSE_REQ <sub>03</sub>	

测试步骤 5:

测试 DUT 在交易阶段收到无效的长时间等待消息后处理是否正确。

1. DUT发送INQUIRY<sub>04</sub> ;
2. iTester在8ms内响应有效的ATI(t)<sub>04</sub> ;
3. iTester在8ms内接收DUT的CONNECT\_REQ<sub>04</sub> ;
4. iTester在8ms内响应CONNECT\_RSP(t)<sub>04</sub> ;
5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>04</sub>或者CHECK2\_REQ(t)<sub>04</sub> ;
6. DUT发送APDATA\_REQ(t2)<sub>04</sub> ;
7. iTester在500ms内响应LTW(e)<sub>04</sub> ;
8. iTester在600ms内接收DUT的消息。

测试流程 5

DUT	发起方测试仪iTester
INQUIRY <sub>04</sub>	ATI(t) <sub>04</sub>
CONNECT_REQ <sub>04</sub>	CONNECT_RSP(t) <sub>04</sub>
CHECK1_REQ(t) <sub>04</sub> or CHECK2_REQ(t) <sub>04</sub>	
APDATA_REQ(t2) <sub>04</sub>	LTW(e) <sub>04</sub>
Mute or INQUIRY or CLOSE_REQ	

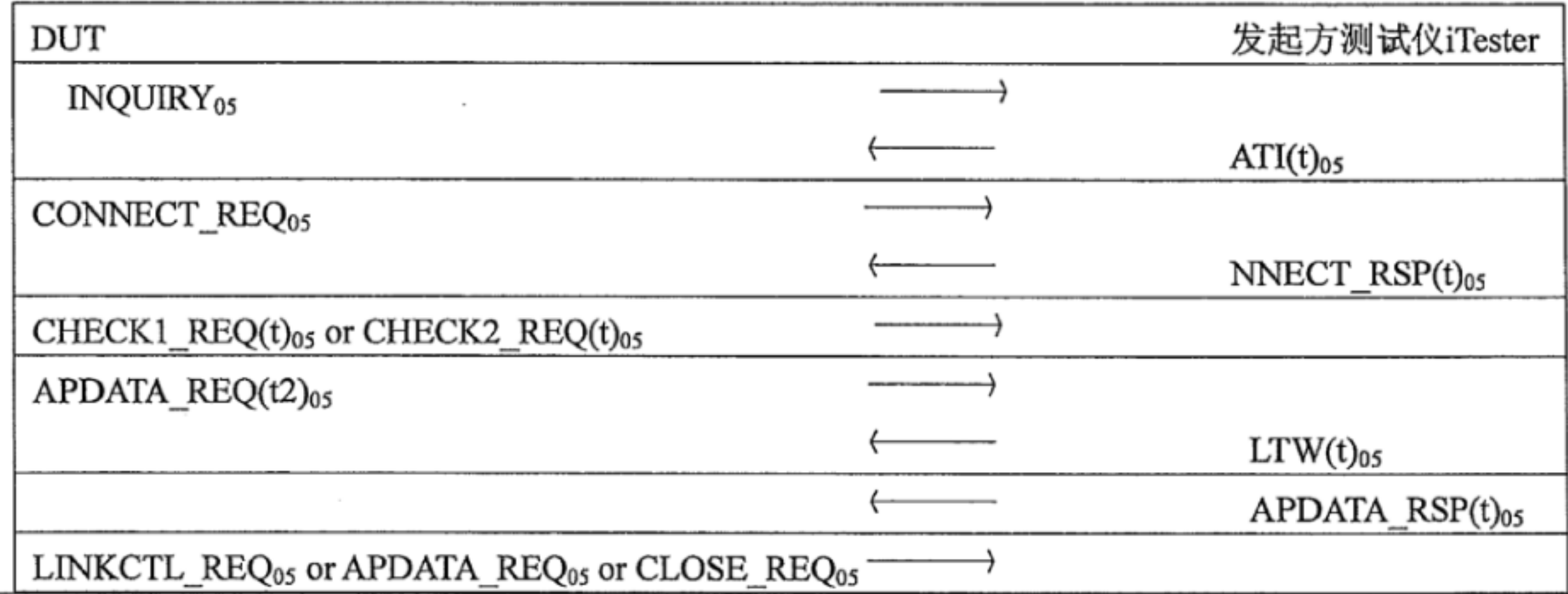
测试步骤 6:

测试 DUT 在交易阶段收到正常的长时间等待消息后处理是否正确。

1. DUT发送INQUIRY<sub>05</sub> ;
2. iTester在8ms内响应有效的ATI(t)<sub>05</sub> ;
3. iTester在8ms内接收DUT的CONNECT\_REQ<sub>04</sub> ;
4. iTester在8ms内响应CONNECT\_RSP(t)<sub>05</sub> ;
5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>05</sub>或者CHECK2\_REQ(t)<sub>05</sub> ;
6. DUT发送APDATA\_REQ(t2)<sub>05</sub> ;

- 7. iTester在500ms内响应LTW(t)<sub>05</sub> ;
- 8. iTester在490ms后发送APDATA\_RSP(t)<sub>05</sub> ;
- 9. iTester在100ms内接收DUT的消息。

测试流程 6



- 预期结果:
- 1. 执行测试步骤1~6, DUT按照测试流程进行响应。
  - 2. APDATA\_REQ(t1)的数据格式符合下表要求:

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0x00或者0xD0h~0xFFh
MsgCode(8bit)	19
MsgLen(16 bit)	24
EncPayLoad	见附录APDATA_REQ(t1)的定义
Checksum(2 byte)	对消息头和消息体数据进行Checksum校验

- 3. APDATA\_REQ(t2)的数据格式符合下表要求:

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0x00或者0xD0h~0xFFh
MsgCode(8bit)	19
MsgLen(16 bit)	248
EncPayLoad	见附录APDATA_REQ(t1)的定义
Checksum(2 byte)	对消息头和消息体数据进行Checksum校验



6.9.4 链路维持测试

测试编号：6.9.4											
测试项目：链路维持测试											
测试目的： 验证发起方DUT在链路维持阶段的数据格式、逻辑操作和响应时间是否符合要求											
预置条件：按照图5进行连接											
测试步骤： 测试步骤 1： 测试 DUT 在链路维持阶段的消息是否正确，在收到无效的维持连接请求后的处理是否正确。 1. DUT发送INQUIRY <sub>00</sub> ； 2. iTester在8ms内响应ATI(t) <sub>00</sub> ； 3. iTester在8ms内接收DUT的请求CONNECT_REQ <sub>00</sub> ； 4. iTester在8ms内响应CONNECT_RSP(t) <sub>00</sub> ； 5. DUT在MC通道持续发送CHECK1_REQ(t) <sub>00</sub> 或者CHECK2_REQ(t) <sub>00</sub> ，iTester检查MC通道消息的数据格式； 6. iTester在100ms内接收DUT发送的LINKCTL_REQ <sub>00</sub> ，并检查LINKCTL_REQ <sub>00</sub> 的数据格式； 7. iTester 在接收到LINKCTL_REQ <sub>00</sub> 后8ms内响应LINKCTL_RSP(e) <sub>00</sub> ； 8. iTester在100ms内接收DUT的命令。 <div>测试流程 1</div> <table><tr><th>DUT</th><th>发起方测试仪iTester</th></tr><tr><td>INQUIRY<sub>00</sub></td><td>ATI(t)<sub>00</sub></td></tr><tr><td>CONNECT_REQ<sub>00</sub></td><td>CONNECT_RSP(t)<sub>00</sub></td></tr><tr><td>CHECK1_REQ or CHECK2_REQ LINKCTL_REQ<sub>00</sub></td><td>LINKCTL_RSP(e)<sub>00</sub></td></tr><tr><td>INQUIRY or Mute or CLOSE</td><td></td></tr></table>		DUT	发起方测试仪iTester	INQUIRY <sub>00</sub>	ATI(t) <sub>00</sub>	CONNECT_REQ <sub>00</sub>	CONNECT_RSP(t) <sub>00</sub>	CHECK1_REQ or CHECK2_REQ LINKCTL_REQ <sub>00</sub>	LINKCTL_RSP(e) <sub>00</sub>	INQUIRY or Mute or CLOSE	
DUT	发起方测试仪iTester										
INQUIRY <sub>00</sub>	ATI(t) <sub>00</sub>										
CONNECT_REQ <sub>00</sub>	CONNECT_RSP(t) <sub>00</sub>										
CHECK1_REQ or CHECK2_REQ LINKCTL_REQ <sub>00</sub>	LINKCTL_RSP(e) <sub>00</sub>										
INQUIRY or Mute or CLOSE											
测试步骤 2： 测试 DUT 在链路维持阶段连续十次接收维持连接响应超时后的处理是否正确。 1. DUT发送INQUIRY <sub>01</sub> ； 2. iTester在8ms内响应ATI(t) <sub>01</sub> ； 3. iTester在8ms内接收DUT的请求CONNECT_REQ <sub>01</sub> ； 4. iTester在8ms内响应CONNECT_RSP(t) <sub>01</sub> ； 5. DUT在MC通道持续发送CHECK1_REQ(t) <sub>01</sub> 或者CHECK2_REQ(t) <sub>01</sub> ； 6. iTester在100ms内接收DUT发送的LINKCTL_REQ <sub>01</sub> ； 7. iTester 在接收到LINKCTL_REQ <sub>01</sub> 后不做响应； 8. 重复9次步骤5~6；											

9. iTester在100ms内接收DUT的命令。

测试流程 2

DUT	发起方测试仪iTester
INQUIRY <sub>01</sub>	ATI(t) <sub>01</sub>
CONNECT_REQ <sub>01</sub>	CONNECT_RSP(t) <sub>01</sub>
CHECK1_REQ or CHECK2_REQ	
LINKCTL_REQ <sub>01</sub>	Mute
.....	.....
LINKCTL_REQ <sub>01</sub>	Mute
INQUIRY or Mute or CLOSE	

预期结果:

- 1. 执行上述测试步骤, DUT按照测试流程进行处理。
- 2. LINKCTL\_REQ<sub>00</sub>的数据格式符合下表要求:

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0x00或者0xD0h~0xFFh
MsgCode(8bit)	22
MsgLen(16 bit)	2
RandData(1 byte)	随机数
Reserved(1 byte)	0
Checksum(2 byte)	对消息头和消息体数据进行Checksum校验

- 3. 步骤2中的MC通道消息应当与下表所描述的CHECK1\_REQ或CHECK2\_REQ保持一致:

CHECK1\_REQ:

字段	值
MsgCode(4bit)	2
MsgLen(4bit)	2
CDC(2 Byte)	取当前响应方生成的随机数IDs的前2字节

CHECK2\_REQ:

字段	值
MsgCode(4bit)	3
MsgLen(4bit)	2
TRI (2 Byte)	取当前响应方生成的随机数IDs的前2字节

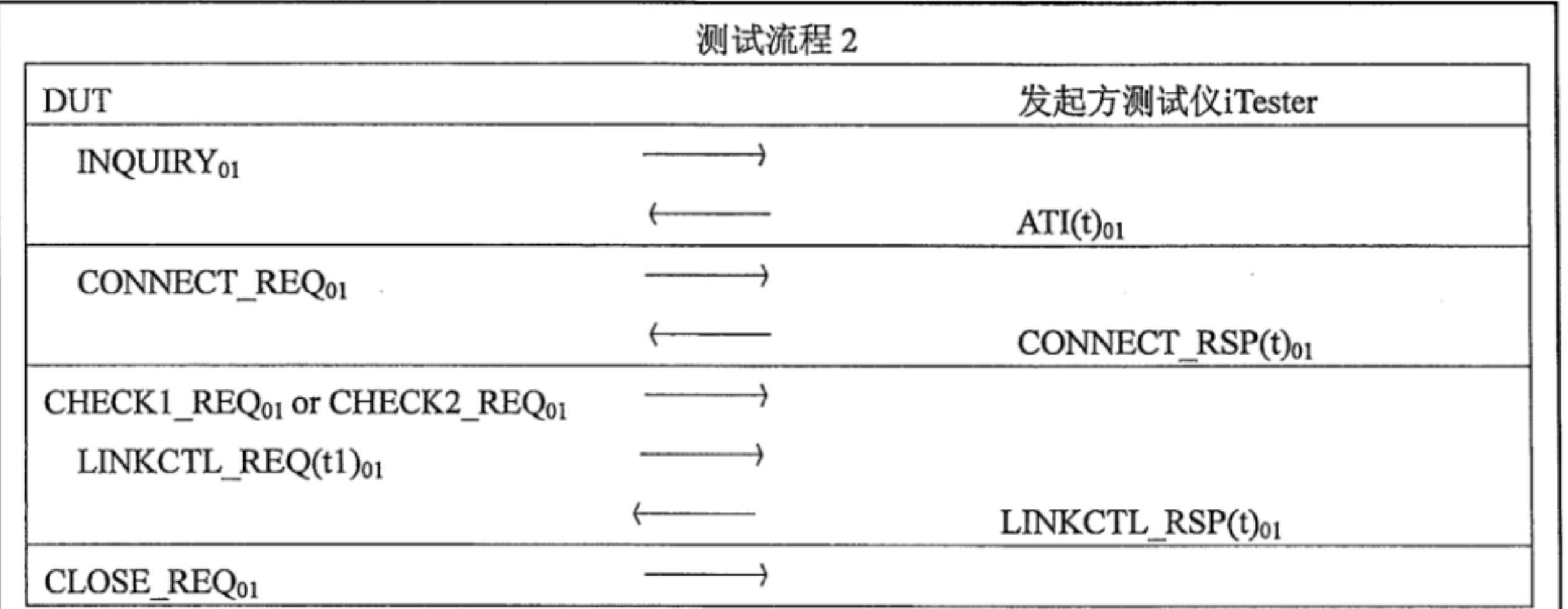
6.9.5 关闭连接测试

测试编号：6.9.5	
测试项目：关闭连接测试	
测试目的： 验证发起方DUT在关闭连接阶段的数据格式、逻辑操作和响应时间是否符合要求	
预置条件：按照图5进行连接	
测试步骤： 测试 DUT 在接收阶段的消息命令是否正确。 1. DUT发送INQUIRY <sub>00</sub> ； 2. iTester在8ms内响应有效的ATI(t) <sub>00</sub> ； 3. DUT发送CONNECT_REQ(t) <sub>00</sub> ； 4. iTester在8ms内响应CONNECT_RSP(t) <sub>00</sub> ； 5. DUT在MC通道持续发送CHECK1_REQ(t) <sub>00</sub> 或者CHECK2_REQ(t) <sub>00</sub> ； 6. DUT发送CLOSE_REQ <sub>00</sub> ， iTester检查CLOSE_REQ <sub>00</sub> 数据格式，进行正确的回应； 7. iTester接收DUT的命令。	
测试流程	
DUT	发起方测试仪iTester
INQUIRY <sub>00</sub> —————>	
	<———— ATI(t) <sub>00</sub>
CONNECT_REQ <sub>00</sub> —————>	
	<———— CONNECT_RSP(t) <sub>00</sub>
CHECK1_REQ(t) <sub>00</sub> or CHECK2_REQ(t) <sub>00</sub> —————>	
CLOSE_REQ <sub>00</sub> —————>	
	<———— Mute or CLOSE_RSP(t) <sub>00</sub>
INQUIRY or Mute —————>	
预期结果：	
1. 执行上述测试步骤，DUT按照测试流程进行处理。	
2. CLOSE_REQ <sub>00</sub> 的数据格式符合下表要求：	
字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0x00或者0xD0h~0xFFh
MsgCode(8bit)	26
MsgLen(16 bit)	4
NeedResp(1 byte)	0或1
Reserved(3 byte )	0
CheckSum(2 byte)	对消息头和消息体数据进行CheckSum校验

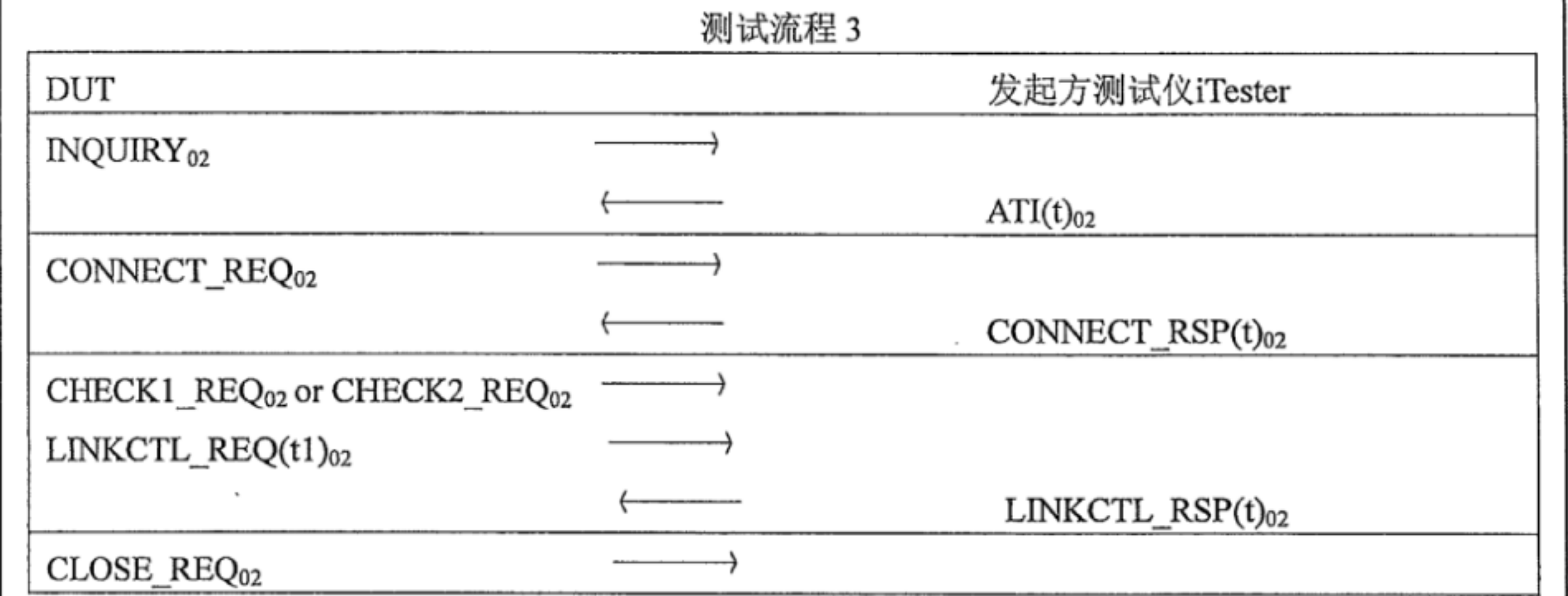
## 6.9.6 异常处理测试

测试编号：6.9.6	
测试项目：异常处理测试	
测试目的： 验证发起方DUT异常处理机制的数据格式、逻辑操作和响应时间是否符合要求	
预置条件：按照图5进行连接	
测试步骤：	
测试步骤 1：	
测试 DUT 在交易阶段收到状态异常（0x01）的维持连接响应消息后处理是否正确。	
1. DUT发送INQUIRY <sub>00</sub> ；	
2. iTester在8ms内响应有效的ATI(t) <sub>00</sub> ；	
3. DUT发送CONNECT_REQ(t) <sub>00</sub> ；	
4. iTester在8ms内响应CONNECT_RSP(t) <sub>00</sub> ；	
5. DUT在MC通道持续发送CHECK1_REQ(t) <sub>00</sub> 或者CHECK2_REQ(t) <sub>00</sub> ；	
6. iTester在100ms内接收DUT的LINKCTL_REQ <sub>00</sub> ；	
7. iTester在8ms内响应LINKCTL_RSP(t) <sub>00</sub> ，其Status字段置为0x01；	
8. iTester在100ms内等待DUT的命令。	
测试流程 1	
DUT	发起方测试仪iTester
INQUIRY <sub>00</sub>	ATI(t) <sub>00</sub>
CONNECT_REQ <sub>00</sub>	CONNECT_RSP(t) <sub>00</sub>
CHECK1_REQ <sub>00</sub> or CHECK2_REQ <sub>00</sub>	
LINKCTL_REQ(t) <sub>00</sub>	LINKCTL_RSP(t) <sub>00</sub>
CLOSE_REQ <sub>00</sub>	
测试步骤 2：	
测试 DUT 在交易阶段收到状态异常（0x02）的维持连接响应消息后处理是否正确。	
1. DUT发送INQUIRY <sub>01</sub> ；	
2. iTester在8ms内响应有效的ATI(t) <sub>01</sub> ；	
3. DUT发送CONNECT_REQ(t) <sub>01</sub> ；	
4. iTester在8ms内响应CONNECT_RSP(t) <sub>01</sub> ；	
5. DUT在MC通道持续发送CHECK1_REQ(t) <sub>01</sub> 或者CHECK2_REQ(t) <sub>01</sub> ；	
6. iTester在100ms内接收DUT的LINKCTL_REQ <sub>01</sub> ；	
7. iTester在8ms内响应LINKCTL_RSP(t) <sub>01</sub> ，其Status字段置为0x02；	
8. iTester在100ms内等待DUT的命令。	





- 测试步骤 3:
- 测试 DUT 在交易阶段收到状态异常 (0x82) 的维持连接响应消息后处理是否正确。
1. DUT发送INQUIRY<sub>02</sub> ;
  2. iTester在8ms内响应有效的ATI(t)<sub>02</sub> ;
  3. DUT发送CONNECT\_REQ(t)<sub>02</sub> ;
  4. iTester在8ms内响应CONNECT\_RSP(t)<sub>02</sub> ;
  5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>02</sub>或者CHECK2\_REQ(t)<sub>02</sub> ;
  6. iTester在100ms内接收DUT的LINKCTL\_REQ<sub>02</sub> ;
  7. iTester在8ms内响应LINKCTL\_RSP(t)<sub>02</sub>, 其Status字段置为0x82;
  8. iTester在100ms内等待DUT的命令。



- 测试步骤 4:
- 测试 DUT 在交易阶段收到状态异常(0x01)的数据交易响应消息后处理是否正确。
1. DUT发送INQUIRY<sub>03</sub> ;
  2. iTester在8ms内响应有效的ATI(t)<sub>03</sub> ;
  3. DUT发送CONNECT\_REQ(t)<sub>03</sub> ;
  4. iTester在8ms内响应CONNECT\_RSP(t)<sub>03</sub> ;
  5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>03</sub>或者CHECK2\_REQ(t)<sub>03</sub> ;

- 6. DUT发送APDATA\_REQ(t1)<sub>03</sub> ;
- 7. iTester在500ms内响应APDATA\_RSP(t)<sub>03</sub>, 其Status字段置为0x01;
- 8. iTester在100ms内接收DUT的命令。

测试流程 4

DUT	发起方测试仪iTester
INQUIRY <sub>03</sub>	ATI(t) <sub>03</sub>
CONNECT_REQ <sub>03</sub>	CONNECT_RSP(t) <sub>03</sub>
CHECK1_REQ <sub>03</sub> or CHECK2_REQ <sub>03</sub>	
APDATA_REQ(t1) <sub>03</sub>	APDATA_RSP(t) <sub>03</sub>
CLOSE_REQ <sub>03</sub>	

测试步骤 5:  
测试 DUT 在交易阶段收到状态异常(0x02)的数据交易响应消息后处理是否正确。

- 1. DUT发送INQUIRY<sub>04</sub> ;
- 2. iTester在8ms内响应有效的ATI(t)<sub>04</sub> ;
- 3. DUT发送CONNECT\_REQ(t)<sub>04</sub> ;
- 4. iTester在8ms内响应CONNECT\_RSP(t)<sub>04</sub> ;
- 5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>04</sub>或者CHECK2\_REQ(t)<sub>04</sub> ;
- 6. DUT发送APDATA\_REQ(t1)<sub>04</sub> ;
- 7. iTester在500ms内响应APDATA\_RSP(t)<sub>04</sub>, 其Status字段置为0x02;
- 8. iTester在100ms内接收DUT的命令。

测试流程 5

DUT	发起方测试仪iTester
INQUIRY <sub>04</sub>	ATI(t) <sub>04</sub>
CONNECT_REQ <sub>04</sub>	CONNECT_RSP(t) <sub>04</sub>
CHECK1_REQ <sub>04</sub> or CHECK2_REQ <sub>04</sub>	
APDATA_REQ(t1) <sub>04</sub>	APDATA_RSP(t) <sub>04</sub>
CLOSE_REQ <sub>04</sub>	

测试步骤 6:  
测试 DUT 在交易阶段收到状态异常(0x82)的数据交易响应消息后处理是否正确。

- 1. DUT发送INQUIRY<sub>05</sub> ;
- 2. iTester在8ms内响应有效的ATI(t)<sub>05</sub> ;
- 3. DUT发送CONNECT\_REQ(t)<sub>05</sub> ;
- 4. iTester在8ms内响应CONNECT\_RSP(t)<sub>05</sub> ;

- 5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>05</sub>或者CHECK2\_REQ(t)<sub>05</sub>；
- 6. DUT发送APDATA\_REQ(t1)<sub>05</sub>；
- 7. iTester在500ms内响应APDATA\_RSP(t)<sub>05</sub>，其Status字段置为0x82；
- 8. iTester在100ms内接收DUT的命令。

测试流程 6

测试步骤 7:

测试 DUT 在交易阶段收到状态异常(0x01)的长时间等待消息后处理是否正确。

- 1. DUT发送INQUIRY<sub>06</sub>；
- 2. iTester在8ms内响应有效的ATI(t)<sub>06</sub>；
- 3. DUT发送CONNECT\_REQ(t)<sub>06</sub>；
- 4. iTester在8ms内响应CONNECT\_RSP(t)<sub>06</sub>；
- 5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>06</sub>或者CHECK2\_REQ(t)<sub>06</sub>；
- 6. DUT发送APDATA\_REQ(t1)<sub>06</sub>；
- 7. iTester在8ms内响应LTW(t)<sub>06</sub>，其Status字段置为0x01；
- 8. iTester在100ms内接收DUT的命令。

测试流程 7

DUT	发起方测试仪iTester
INQUIRY <sub>06</sub>	ATI(t) <sub>06</sub>
CONNECT_REQ <sub>06</sub>	CONNECT_RSP(t) <sub>06</sub>
CHECK1_REQ <sub>06</sub> or CHECK2_REQ <sub>06</sub>	
APDATA_REQ(t1) <sub>06</sub>	LTW(t) <sub>06</sub>
CLOSE_REQ <sub>06</sub>	

测试步骤 8:

测试 DUT 在交易阶段收到状态异常(0x02)的长时间等待消息后处理是否正确。

- 1. DUT发送INQUIRY<sub>07</sub>；
- 2. iTester在8ms内响应有效的ATI(t)<sub>07</sub>；
- 3. DUT发送CONNECT\_REQ(t)<sub>07</sub>；
- 4. iTester在8ms内响应CONNECT\_RSP(t)<sub>07</sub>；
- 5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>07</sub>或者CHECK2\_REQ(t)<sub>07</sub>；
- 6. DUT发送APDATA\_REQ(t1)<sub>07</sub>；
- 7. iTester在8ms内响应LTW(t)<sub>07</sub>，其Status字段置为0x02；
- 8. iTester在100ms内接收DUT的命令。

测试流程 8

DUT	发起方测试仪iTester
INQUIRY <sub>07</sub>	ATI(t) <sub>07</sub>
CONNECT_REQ <sub>07</sub>	CONNECT_RSP(t) <sub>07</sub>
CHECK1_REQ <sub>07</sub> or CHECK2_REQ <sub>07</sub>	
APDATA_REQ(t1) <sub>07</sub>	LTW(t) <sub>07</sub>
CLOSE_REQ <sub>07</sub>	

- 测试步骤 9:
- 测试 DUT 在交易阶段收到状态异常(0x82)的长时间等待消息后处理是否正确。
1. DUT发送INQUIRY<sub>08</sub> ;
  2. iTester在8ms内响应有效的ATI(t)<sub>08</sub> ;
  3. DUT发送CONNECT\_REQ(t)<sub>08</sub> ;
  4. iTester在8ms内响应CONNECT\_RSP(t)<sub>08</sub> ;
  5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>08</sub>或者CHECK2\_REQ(t)<sub>08</sub> ;
  6. DUT发送APDATA\_REQ(t1)<sub>08</sub> ;
  7. iTester在8ms内响应LTW(t)<sub>08</sub>, 其Status字段置为0x82;
  8. iTester在100ms内接收DUT的命令。

测试流程 9

DUT	发起方测试仪iTester
INQUIRY <sub>08</sub>	ATI(t) <sub>08</sub>
CONNECT_REQ <sub>08</sub>	CONNECT_RSP(t) <sub>08</sub>
CHECK1_REQ <sub>08</sub> or CHECK2_REQ <sub>08</sub>	
APDATA_REQ(t1) <sub>08</sub>	LTW(t) <sub>08</sub>
CLOSE_REQ <sub>08</sub>	

- 测试步骤 10:
- 测试 DUT 在交易阶段收到冲突检测消息后处理是否正确。
1. DUT发送INQUIRY<sub>09</sub> ;
  2. iTester在8ms内响应有效的ATI(t)<sub>09</sub> ;
  3. DUT发送CONNECT\_REQ(t)<sub>09</sub> ;
  4. iTester在8ms内响应CONNECT\_RSP(t)<sub>09</sub> ;
  5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>09</sub>或者CHECK2\_REQ(t)<sub>09</sub>, iTester接收磁通道消息, 若收到



MC通道的消息为CHECK1\_REQ<sub>09</sub>则执行进入步骤6，若为CHECK2\_REQ<sub>09</sub>则测试Pass且结束；  
6. iTeser每4ms内随机的发送CHECK\_RSP(t1)<sub>09</sub>，直至DUT收到此消息；  
7. iTeser在100ms内接收DUT的命令。

测试流程 10

DUT		发起方测试仪iTeser
INQUIRY <sub>09</sub>	————>	
	<————	ATI(t) <sub>09</sub>
CONNECT_REQ <sub>09</sub>	————>	
	<————	CONNECT_RSP(t) <sub>09</sub>
CHECK1_REQ <sub>09</sub> or CHECK2_REQ <sub>09</sub>	————>  ————>	
	<————	CHECK_RSP(t1) <sub>09</sub>
CLOSE_REQ <sub>09</sub>	————>	

预期结果：  
执行测试步骤1~10，DUT的响应与测试流程保持一致

6.9.7 超时时间测试

测试编号：6.9.7															
测试项目：超时时间测试															
测试目的： 验证发起方DUT在激活阶段、连接阶段、链路维持阶段、数据交易阶段的超时时间是否符合要求															
预置条件：按照图5进行连接															
测试步骤： 测试步骤 1： 测试 DUT 在激活阶段的超时时间。 1. DUT发送INQUIRY <sub>00</sub> ； 2. iTester在收到INQUIRY后的第6ms响应有效的ATI(t) <sub>00</sub> ； 3. iTester在8ms内接收DUT响应，若收到CONNECT_REQ <sub>00</sub> 则测试Pass,否则测试Fail； 4. DUT发送INQUIRY <sub>00</sub> ； 5. iTester在收到INQUIRY <sub>00</sub> 时立即启动计时，第10ms响应有效的ATI(t) <sub>00</sub> ； 6. iTester在8ms内接收DUT响应，若收到CONNECT_REQ <sub>00</sub> 则测试Fail，否则测试Pass。 <div>测试流程 1</div> <table><tr><th>DUT</th><th>发起方测试仪iTester</th></tr><tr><td>INQUIRY<sub>00</sub></td><td>————→</td></tr><tr><td></td><td>←———— ATI(t)<sub>00</sub></td></tr><tr><td>CONNECT_REQ<sub>00</sub></td><td>————→</td></tr><tr><td>INQUIRY<sub>00</sub></td><td>————→</td></tr><tr><td></td><td>←———— ATI(t)<sub>00</sub></td></tr><tr><td>Mute or INQUIRY</td><td>————→</td></tr></table>		DUT	发起方测试仪iTester	INQUIRY <sub>00</sub>	————→		←———— ATI(t) <sub>00</sub>	CONNECT_REQ <sub>00</sub>	————→	INQUIRY <sub>00</sub>	————→		←———— ATI(t) <sub>00</sub>	Mute or INQUIRY	————→
DUT	发起方测试仪iTester														
INQUIRY <sub>00</sub>	————→														
	←———— ATI(t) <sub>00</sub>														
CONNECT_REQ <sub>00</sub>	————→														
INQUIRY <sub>00</sub>	————→														
	←———— ATI(t) <sub>00</sub>														
Mute or INQUIRY	————→														
测试步骤 2： 测试 DUT 在连接阶段的超时时间。 1. DUT发送INQUIRY <sub>01</sub> ； 2. iTester在8ms内响应ATI(t) <sub>01</sub> ； 3. iTester在8ms内接收CONNECT_REQ <sub>01</sub> ； 4. iTester在收到CONNECT_REQ <sub>01</sub> 时立即启动计时，第6ms发送CONNECT_RSP(t) <sub>01</sub> ； 5. Tester在100ms内接收LINKCTL_REQ <sub>01</sub> ，若收到LINKCTL_REQ <sub>01</sub> 则测试Pass，否则测试Fail； 6. DUT发送INQUIRY <sub>01</sub> ； 7. iTester在8ms内响应ATI(t) <sub>01</sub> ； 8. iTester在8ms内接收CONNECT_REQ <sub>01</sub> ； 9. iTester在收到CONNECT_REQ <sub>01</sub> 后第10ms发送CONNECT_RSP(t) <sub>01</sub> ； 10. iTester在100ms内接收DUT响应，若收到LINKCTL_REQ <sub>01</sub> 则测试Fail，否则测试Pass。															

测试流程 2		
DUT		发起方测试仪iTester
INQUIRY <sub>01</sub>	————→	
	←————	ATI(t) <sub>01</sub>
CONNECT_REQ <sub>01</sub>	————→	
	←————	CONNECT_RSP(t) <sub>01</sub>
LINKCTL_REQ <sub>01</sub>	————→	
INQUIRY <sub>01</sub>	————→	
	←————	ATI(t) <sub>01</sub>
CONNECT_REQ <sub>01</sub>	————→	
	←————	CONNECT_RSP(t) <sub>01</sub>
Mute or INQUIRY	————→	

测试步骤 3:

测试 DUT 接收维持连接响应的超时时间。

1. DUT发送INQUIRY<sub>02</sub> ;
2. iTester响应ATI(t)<sub>02</sub> ;
3. iTester在8ms内接收CONNECT\_REQ<sub>02</sub> ;
4. iTester发送CONNECT\_RSP(t)<sub>02</sub> ;
5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>02</sub>或者CHECK2\_REQ(t)<sub>02</sub> ;
6. iTester在100ms内接收LINKCTL\_REQ<sub>02</sub>, 在收到LINKCTL\_REQ<sub>02</sub>时立即启动计时, 第10ms回应LINKCTL\_RSP(t)<sub>02</sub> ;
7. 重复9次执行测试步骤6;
8. iTester在100ms内接收DUT的命令。

测试流程 3		
DUT		发起方测试仪iTester
INQUIRY <sub>02</sub>	————→	
	←————	ATI(t) <sub>02</sub>
CONNECT_REQ <sub>02</sub>	————→	
	←————	CONNECT_RSP(t) <sub>02</sub>
CHECK1_REQ <sub>02</sub> or CHECK2_REQ <sub>02</sub>	————→	
LINKCTL_REQ <sub>02</sub>	————→	
	←————	LINKCTL_RSP(t) <sub>02</sub>
LINKCTL_REQ <sub>02</sub>	————→	
	←————	LINKCTL_RSP(t) <sub>02</sub>
.....	.....	.....
Mute or INQUIRY	————→	

测试步骤 4:

测试 DUT 接收数据交换响应的超时时间。

- 1. DUT发送INQUIRY<sub>03</sub> ;
- 2. iTester响应ATI(t)<sub>03</sub> ;
- 3. iTester在8ms内接收CONNECT\_REQ<sub>03</sub> ;
- 4. iTester在发送CONNECT\_RSP(t)<sub>03</sub> ;
- 5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>03</sub>或者CHECK2\_REQ(t)<sub>03</sub> ;
- 6. DUT发送APDATA\_REQ(t1)<sub>03</sub> ;
- 7. iTester在收到APDATA\_REQ(t1)<sub>03</sub>时立即启动计时, 第400ms响应APDATA\_RSP(t)<sub>03</sub> ;
- 8. iTester在100ms内接收DUT的命令, 若收到LINKCTL\_REQ<sub>03</sub> or APDATA\_REQ<sub>03</sub> or CLOSE\_REQ<sub>03</sub>则测试Pass, 否则测试Fail。

测试流程 4

DUT	发起方测试仪iTester
INQUIRY <sub>03</sub>	ATI(t) <sub>03</sub>
CONNECT_REQ <sub>03</sub>	CONNECT_RSP(t) <sub>03</sub>
CHECK1_REQ <sub>03</sub> or CHECK2_REQ <sub>03</sub> APTATA_REQ(t1) <sub>03</sub>	APDATA_RSP(t) <sub>03</sub>
LINKCTL_REQ <sub>03</sub> or APDATA_REQ <sub>03</sub> or CLOSE_REQ <sub>03</sub>	

测试步骤 5:

测试 DUT 接收数据交换响应的超时时间。

- 1. DUT发送INQUIRY<sub>04</sub> ;
- 2. iTester响应ATI(t)<sub>04</sub> ;
- 3. iTester在8ms内接收CONNECT\_REQ<sub>04</sub> ;
- 4. iTester在收到CONNECT\_REQ<sub>04</sub>后发送CONNECT\_RSP(t)<sub>04</sub> ;
- 5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>04</sub>或者CHECK2\_REQ(t)<sub>04</sub> ;
- 6. DUT发送APDATA\_REQ(t1)<sub>04</sub> ;
- 7. iTester在收到APDATA\_REQ(t1)<sub>04</sub>时立即启动计时, 第600ms响应APDATA\_RSP(t)<sub>04</sub> ;
- 8. iTester在100ms内接收DUT消息, 若收到LINKCTL\_REQ<sub>04</sub>则测试Fail且结束, 否则进行下一步。

测试流程 5

DUT	发起方测试仪iTester
INQUIRY <sub>04</sub>	ATI(t) <sub>04</sub>
CONNECT_REQ <sub>04</sub>	CONNECT_RSP(t) <sub>04</sub>



CHECK1_REQ <sub>04</sub> or CHECK2_REQ <sub>04</sub>	————→	
APTATA_REQ(t1) <sub>04</sub>	————→	
	←————	APDATA_RSP(t) <sub>04</sub>
Mute or INQUIRY	————→	

测试步骤 6:

测试 DUT 接收数据交换响应 LTW 的超时时间。

1. DUT发送INQUIRY<sub>05</sub> ;

2. iTester响应ATI(t)<sub>05</sub> ;

3. iTester在8ms内接收CONNECT\_REQ<sub>05</sub> ;

4. iTester在收到CONNECT\_REQ<sub>05</sub>后发送CONNECT\_REQ<sub>05</sub> ;

5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>05</sub>或者CHECK2\_REQ(t)<sub>05</sub> ;

6. DUT发送APDATA\_REQ(t1)<sub>05</sub> ;

7. iTester在收到APDATA\_REQ(t1)<sub>05</sub>时立即启动计时, 第400ms响应LTW(t)<sub>05</sub> ;

8. iTester在发送完LTW(t)<sub>05</sub>之后400ms内发送APDATA\_RSP(t)<sub>05</sub> ;

9. Tester在在100ms内接收DUT的命令, 若收到LINKCTL\_REQ<sub>05</sub> or APDATA\_REQ<sub>05</sub> or CLOSE\_REQ<sub>05</sub>则测试Pass, 否则测试Fail。

测试流程 6

DUT		发起方测试仪iTester
INQUIRY <sub>05</sub>	————→	
	←————	ATI(t) <sub>05</sub>
CONNECT_REQ <sub>05</sub>	————→	
	←————	CONNECT_RSP(t) <sub>05</sub>
CHECK1_REQ <sub>05</sub> or CHECK2_REQ <sub>05</sub>	————→	
APTATA_REQ(t1) <sub>05</sub>	————→	
	←————	LTW(t) <sub>05</sub>
LINKCTL_REQ <sub>05</sub> or APDATA_REQ <sub>05</sub> or CLOSE_REQ <sub>05</sub>	————→	

测试步骤 7:

测试 DUT 接收数据交换响应 LTW 的超时时间。

1. DUT发送INQUIRY<sub>06</sub> ;

2. iTester响应ATI(t)<sub>06</sub> ;

3. iTester在8ms内接收CONNECT\_REQ<sub>06</sub> ;

4. iTester在收到CONNECT\_REQ<sub>06</sub>后发送CONNECT\_REQ<sub>06</sub> ;

5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>06</sub>或者CHECK2\_REQ(t)<sub>06</sub> ;

6. DUT发送APDATA\_REQ(t1)<sub>06</sub> ;

7. iTester在收到APDATA\_REQ(t1)<sub>06</sub>时立即启动计时, 第600ms响应LTW(t)<sub>05</sub> ;

8. iTester在100ms内接收LINKCTL\_REQ<sub>06</sub>, 若收到LINKCTL\_REQ<sub>06</sub>则测试Fail且结束, 否则测试Pass。

测试流程 7		
DUT	发起方测试仪iTester	
INQUIRY <sub>06</sub>	————→	
	←————	ATI(t) <sub>06</sub>
CONNECT_REQ <sub>06</sub>	————→	
	←————	CONNECT_RSP(t) <sub>06</sub>
CHECK1_REQ <sub>06</sub> or CHECK2_REQ <sub>06</sub>	————→	
APTATA_REQ(t1) <sub>06</sub>	————→	
	←————	LTW(t) <sub>06</sub>
Mute or INQUIRY	————→	

测试步骤 8:

测试 DUT 在维持连接阶段发送维持连接请求的时间间隔是否符合 YD/T XXX.10。

1. DUT发送INQUIRY<sub>07</sub> ；
2. iTester响应ATI(t)<sub>07</sub> ；
3. iTester在8ms内接收CONNECT\_REQ<sub>07</sub> ；
4. iTester在8ms内响应CONNECT\_RSP<sub>07</sub> ；
5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>07</sub>或者CHECK2\_REQ(t)<sub>07</sub> ；
6. iTester在100ms内接收LINKCTL\_REQ<sub>07</sub>，并响应LINKCTL\_RSP<sub>07</sub> ；
7. 重复20次执行步骤6；
8. iTester计算相邻两次收到LINKCTL\_REQ<sub>07</sub>的时间间隔，若为为44ms则测试Pass，否则Fail。

测试流程 8		
DUT	发起方测试仪iTester	
INQUIRY <sub>07</sub>	————→	
	←————	ATI(t) <sub>07</sub>
CONNECT_REQ <sub>07</sub>	————→	
	←————	CONNECT_RSP(t) <sub>07</sub>
CHECK1_REQ <sub>07</sub> or CHECK2_REQ <sub>07</sub>	————→	
LINKCTL_REQ(t) <sub>07</sub>	————→	
	←————	LINKCTL_RSP <sub>07</sub>
.....	.....	.....

预期结果:

1. 执行测试步骤1~8，DUT按照测试流程进行响应。
2. 测试过程中不出现测试Fail情况

## 6.9.8 加密算法验证

测试编号: 6.9.8	
测试项目: 加密算法验证	
测试目的: 验证发起方DUT支持的加密算法是否符合要求	
预置条件: 按照图5进行连接	
测试步骤: 1. DUT发送INQUIRY <sub>00</sub> ; 2. iTester再响应ATI <sub>00</sub> ; 3. iTester在8ms内接收CONNECT_REQ <sub>00</sub> ; 4. iTester在收到CONNECT_REQ <sub>00</sub> 后根据 'EncAlg' 的值查找下表, 执行对应步骤进行测试, 若值不在下表中则直接测试Fail。	
'EncAlg'	执行测试步骤
0100	步骤1
0200	步骤2
0300	步骤1、2
0400	步骤3
0500	步骤1、3
0600	步骤2、3
0700	步骤1、2、3
0800	步骤4
0900	步骤1、4
0A00	步骤2、4
0B00	步骤1、2、4
0C00	步骤3、4
0D00	步骤1、3、4
0E00	步骤2、3、4
0F00	步骤1、2、3、4
1000	步骤5
1100	步骤1、5
1200	步骤2、5
1300	步骤1、2、5
1400	步骤3、5
1500	步骤1、3、5
1600	步骤2、3、5
1700	步骤1、2、3、5
1800	步骤4、5
1900	步骤1、4、5
1A00	步骤2、4、5

1B00	步骤1、2、4、5
1C00	步骤3、4、5
1D00	步骤1、3、4、5
1E00	步骤2、3、4、5
1F00	步骤1、2、3、4、5
2000	步骤6
2100	步骤1、6
2200	步骤2、6
2300	步骤1、2、6
2400	步骤3、6
2500	步骤1、3、6
2600	步骤2、3、6
2700	步骤1、2、3、6
2800	步骤4、6
2900	步骤1、4、6
2A00	步骤2、4、6
2B00	步骤1、2、4、6
2C00	步骤3、4、6
2D00	步骤1、3、4、6
2E00	步骤2、3、4、6
2F00	步骤1、2、3、4、6
3000	步骤5、6
3100	步骤1、5、6
3200	步骤2、5、6
3300	步骤1、2、5、6
3400	步骤3、5、6
3500	步骤1、3、5、6
3600	步骤2、3、5、6
3700	步骤1、2、3、5、6
3800	步骤4、5、6
3900	步骤1、4、5、6
3A00	步骤2、4、5、6
3B00	步骤1、2、4、5、6
3C00	步骤3、4、5、6
3D00	步骤1、3、4、5、6
3E00	步骤2、3、4、5、6
3F00	步骤1、2、3、4、5、6



测试步骤1:

测试 DUT 的 3DES-EBC 算法。

- 1. 在DUT的工作区域内放置发起方测试仪iTester;
- 2. DUT发送INQUIRY<sub>00</sub> ;
- 3. iTester响应ATI<sub>00</sub> ;
- 4. iTester在8ms内接收CONNECT\_REQ<sub>00</sub> ;
- 5. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>00</sub>或者CHECK2\_REQ(t)<sub>00</sub> ;
- 6. iTester在8ms内响应CONNECT\_RSP(t)<sub>00</sub>, ‘EncAlg’ 的值为0x0100;
- 7. DUT发送APDATA\_REQ(t1)<sub>00</sub>, 其EncPayload字段的明文数据已知;
- 8. iTester接收ADATA\_REQ(t1)<sub>00</sub>, 进行加密算法验证。

测试流程 1

DUT	发起方测试仪iTester
INQUIRY <sub>00</sub>	ATI(t) <sub>00</sub>
CONNECT_REQ <sub>00</sub>	CONNECT_RSP(t) <sub>00</sub>
CHECK1_REQ(t) <sub>00</sub> or CHECK2_REQ(t) <sub>00</sub>	
APDATA_REQ(t1) <sub>00</sub>	

测试步骤 2:

测试 DUT 的 3DES-ECB 算法。

- 1. 在DUT的工作区域内放置发起方测试仪iTester;
- 2. DUT发送INQUIRY<sub>01</sub> ;
- 3. iTester响应ATI<sub>01</sub> ;
- 4. iTester在8ms内接收CONNECT\_REQ<sub>01</sub> ;
- 5. iTester在8ms内响应CONNECT\_RSP(t)<sub>01</sub>, ‘EncAlg’ 的值为0x0200;
- 6. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>00</sub>或者CHECK2\_REQ(t)<sub>00</sub> ;
- 7. DUT发送APDATA\_REQ(t1)<sub>01</sub>, 其EncPayload字段的明文数据已知;
- 8. iTester接收ADATA\_REQ(t1)<sub>01</sub>, 进行加密算法验证。

测试流程 2

DUT	发起方测试仪iTester
INQUIRY <sub>01</sub>	ATI(t) <sub>01</sub>
CONNECT_REQ <sub>01</sub>	CONNECT_RSP(t) <sub>01</sub>
CHECK1_REQ(t) <sub>00</sub> or CHECK2_REQ(t) <sub>01</sub>	
APDATA_REQ(t1) <sub>01</sub>	

## 测试步骤 3:

测试 DUT 的 AES-ECB 算法。

1. 在DUT的工作区域内放置发起方测试仪iTester;
2. DUT发送INQUIRY<sub>02</sub> ;
3. iTester响应ATI<sub>02</sub> ;
4. iTester在8ms内接收CONNECT\_REQ<sub>02</sub> ;
5. iTester在8ms内响应CONNECT\_RSP(t)<sub>02</sub>, 'EncAlg' 的值为0x0400;
6. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>02</sub>或者CHECK2\_REQ(t)<sub>02</sub> ;
7. DUT发送APDATA\_REQ(t1)<sub>02</sub>, 其EncPayload字段的明文数据已知;
8. iTester接收ADATA\_REQ(t1)<sub>02</sub>, 进行加密算法验证。

测试流程 3

DUT	发起方测试仪iTester
INQUIRY <sub>02</sub>	→
	← ATI(t) <sub>02</sub>
CONNECT_REQ <sub>02</sub>	→
	← CONNECT_RSP(t) <sub>02</sub>
CHECK1_REQ(t) <sub>02</sub> or CHECK2_REQ(t) <sub>02</sub>	→
APDATA_REQ(t1) <sub>02</sub>	→

## 测试步骤 4:

测试 DUT 的 AES-CBC 算法。

1. 在DUT的工作区域内放置发起方测试仪iTester;
2. DUT发送INQUIRY<sub>03</sub> ;
3. iTester响应ATI<sub>03</sub> ;
4. iTester在8ms内接收CONNECT\_REQ<sub>03</sub> ;
5. iTester在8ms内响应CONNECT\_RSP(t)<sub>03</sub>, 'EncAlg' 的值为0x0800;
6. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>03</sub>或者CHECK2\_REQ(t)<sub>03</sub> ;
7. DUT发送APDATA\_REQ(t1)<sub>03</sub>, 其EncPayload字段的明文数据已知;
8. iTester接收ADATA\_REQ(t1)<sub>03</sub>, 进行加密算法验证。

测试流程 4

DUT	发起方测试仪iTester
INQUIRY <sub>03</sub>	→
	← ATI(t) <sub>03</sub>
CONNECT_REQ <sub>03</sub>	→
	← CONNECT_RSP(t) <sub>03</sub>
CHECK1_REQ(t) <sub>03</sub> or CHECK2_REQ(t) <sub>03</sub>	→
APDATA_REQ(t1) <sub>03</sub>	→

测试步骤 5:

测试 DUT 的 SM4-ECB 算法。

- 1. 在DUT的工作区域内放置发起方测试仪iTester;
- 2. DUT发送INQUIRY<sub>04</sub> ;
- 3. iTester响应ATI<sub>04</sub> ;
- 4. iTester在8ms内接收CONNECT\_REQ<sub>04</sub> ;
- 5. iTester在8ms内响应CONNECT\_RSP(t)<sub>04</sub>, ‘EncAlg’ 的值为0x1000;
- 6. DUT在MC通道持续发送CHECK1\_REQ(t)<sub>04</sub>或者CHECK2\_REQ(t)<sub>04</sub> ;
- 7. DUT发送APDATA\_REQ(t1)<sub>04</sub>, 其EncPayload字段的明文数据已知;
- 8. iTester接收ADATA\_REQ(t1)<sub>04</sub>, 进行加密算法验证。

测试流程 5

DUT	发起方测试仪iTester
INQUIRY <sub>04</sub>	ATI(t) <sub>04</sub>
CONNECT_REQ <sub>04</sub>	CONNECT_RSP(t) <sub>04</sub>
CHECK1_REQ(t) <sub>04</sub> or CHECK2_REQ(t) <sub>04</sub>	
APDATA_REQ(t1) <sub>04</sub>	

测试步骤 6:

测试 DUT 的 SM4-CBC 算法。

- 1. 在 DUT 的工作区域内放置发起方测试仪 iTester;
- 2. DUT 发送 INQUIRY<sub>05</sub> ;
- 3. iTester 响应 ATI<sub>05</sub> ;
- 4. iTester 在 8ms 内接收 CONNECT\_REQ<sub>05</sub> ;
- 5. iTester 在 8ms 内响应 CONNECT\_RSP(t)<sub>05</sub>, ‘EncAlg’ 的值为 0x2000;
- 6. DUT 在 MC 通道持续发送 CHECK1\_REQ(t)<sub>05</sub> 或者 CHECK2\_REQ(t)<sub>05</sub> ;
- 7. DUT 发送 APDATA\_REQ(t1)<sub>05</sub>, 其 EncPayload 字段的明文数据已知;
- 8. iTester 接收 ADATA\_REQ(t1)<sub>05</sub>, 进行加密算法验证。

测试流程 6

DUT	发起方测试仪iTester
INQUIRY <sub>05</sub>	ATI(t) <sub>05</sub>
CONNECT_REQ <sub>05</sub>	CONNECT_RSP(t) <sub>05</sub>
CHECK1_REQ(t) <sub>05</sub> or CHECK2_REQ(t) <sub>05</sub>	
APDATA_REQ(t1) <sub>05</sub>	

预期结果:

- 1. 执行测试步骤1~6, 过程中不出现Fail情况, DUT按照测试流程进行响应。
- 2. 执行测试步骤1~6, 过程中所有的算法验证通过



6.9.9 频点范围测试

测试编号：6.9.9											
测试项目：频点范围测试											
测试目的： 验证发起方DUT的频点范围是否符合要求											
预置条件：按照图5进行连接											
测试步骤： 测试步骤 1： 1. DUT发送INQUIRY <sub>00</sub> ； 2. iTester响应ATI <sub>00</sub> ， 设置‘IDs’字段使得射频工作频点为2401Mhz； 3. iTester在8ms内接收CONNECT_REQ <sub>00</sub> ， 如果在频点2401Mhz收到CONNECT_REQ <sub>00</sub> 则进行下一步， 否则测试Fail且结束； 4. iTester在8ms内响应CONNECT_RSP(t) <sub>00</sub> ； 5. iTester在100ms内接收DUT消息； 6. 重复63次执行步骤2~5， 射频工作频点增加1Mhz， 当第63次执行完成时测试Pass， 否则测试Fail。											
测试流程 1											
<table><tr><th>DUT</th><th>发起方测试仪iTester</th></tr><tr><td>INQUIRY<sub>00</sub></td><td>ATI(t)<sub>00</sub></td></tr><tr><td>CONNECT_REQ<sub>00</sub></td><td>CONNECT_RSP(t)<sub>00</sub></td></tr><tr><td>LINKCTL_REQ<sub>00</sub> or APDATA_REQ<sub>00</sub></td><td></td></tr><tr><td>.....</td><td>.....</td></tr></table>		DUT	发起方测试仪iTester	INQUIRY <sub>00</sub>	ATI(t) <sub>00</sub>	CONNECT_REQ <sub>00</sub>	CONNECT_RSP(t) <sub>00</sub>	LINKCTL_REQ <sub>00</sub> or APDATA_REQ <sub>00</sub>		.....	.....
DUT	发起方测试仪iTester										
INQUIRY <sub>00</sub>	ATI(t) <sub>00</sub>										
CONNECT_REQ <sub>00</sub>	CONNECT_RSP(t) <sub>00</sub>										
LINKCTL_REQ <sub>00</sub> or APDATA_REQ <sub>00</sub>											
.....	.....										
测试步骤 2： 如果 DUT 在维持连接阶段磁通道发送的消息为 CHECK1_REQ 则进行该测试步骤， 否则不进行该测试步骤。 1. DUT发送INQUIRY <sub>01</sub> ； 2. iTester在8ms内响应ATI(t) <sub>01</sub> ， 设置‘IDs’的前两个字节为‘00 01’使得CHECK_RSP <sub>01</sub> 的频点为2465Mhz； 3. iTester在8ms内接收CONNECT_REQ <sub>01</sub> ； 4. iTester在收到CONNECT_REQ <sub>01</sub> 后发送CONNECT_RSP(t) <sub>01</sub> ； 5. iTester持续发送22ms CHECK_RSP <sub>01</sub> ， 若发送成功则测试Pass， 否则测试Fail； 6. 重复3次执行步骤1~5， 分别设置ATI的 ‘IDs’ 前两个字节为 ‘00 02’、 ‘00 03’ 和 ‘00 04’， 使得对应的CHECK_RSP <sub>01</sub> 频点为2466Mhz、 2467Mhz和2468Mhz。											
测试流程 2											
<table><tr><th>DUT</th><th>发起方测试仪iTester</th></tr><tr><td>INQUIRY<sub>01</sub></td><td>ATI(t)<sub>01</sub></td></tr><tr><td>CONNECT_REQ<sub>01</sub></td><td>CONNECT_RSP(t)<sub>01</sub></td></tr><tr><td></td><td>CHECK_RSP<sub>01</sub></td></tr></table>		DUT	发起方测试仪iTester	INQUIRY <sub>01</sub>	ATI(t) <sub>01</sub>	CONNECT_REQ <sub>01</sub>	CONNECT_RSP(t) <sub>01</sub>		CHECK_RSP <sub>01</sub>		
DUT	发起方测试仪iTester										
INQUIRY <sub>01</sub>	ATI(t) <sub>01</sub>										
CONNECT_REQ <sub>01</sub>	CONNECT_RSP(t) <sub>01</sub>										
	CHECK_RSP <sub>01</sub>										
预期结果： 1. 执行测试步骤1和2， DUT按照测试流程进行响应。 2. 测试过程中不出现Fail											



附 录 A  
(规范性附录)  
测试命令

本附录定义了协议测试中使用的有效测试命令和无效测试命令，其中有效测试命令格式和内容均符合《手机支付 基于2.45GHz RCC（限域通信）技术的非接触射频接口技术要求》第10.3节的规定，无效测试命令的格式与有效测试命令相同，但其部分字段内容的值与有效测试命令不同。

A.1 激活请求

有效激活命令 INQUIRY(t)定义如下表：

字段	值
MsgCode(4 bit)	0
MsgLen(4 bit)	15
Rfu(4bit)	0
InitiatorVersion(4bit))	3
IDm(14 byte)	随机数

无效激活命令 INQUIRY(e)的‘MsgCode’值与上表不同，或者‘MsgLen’或‘InitiatorVersion’值与上表不同，其余内容与上表一致。

A.2 激活响应

有效的激活响应 ATI(t)定义如下表：

字段	值
Rfu (4 bit)	0
FormatType(4 bit)	8
Status(1 Byte)	0
MsgCode(1 Byte)	16
MsgLen(2 Byte)	24
IDs(5 byte)	随机数
TargetID(8 Byte)	响应方标识（自定义）
AccessVersion (1 byte)	0x03
Mac(4 Byte)	响应方使用K0作为密钥对（IDs  TargetID  AccessVersion）进行MAC计算。16字节K0使用SKG0中同样的扩展方式由14字节IDm扩展生成。 MAC算法见YD/T XXX.10的附录B.5
Reserved (6 Byte)	0
Checksum(2 Byte)	对消息头和消息体数据进行Checksum校验

无效的激活命令 ATI(e)的‘Mac’值与正确值不同，其余内容与上表一致。

## A.3 连接请求

有效的连接请求 CONNECT\_REQ(t)的定义如下:

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0
MsgCode(8bit)	17
MsgLen(16 bit)	24
InitiatorType(1 byte)	'A'
InitiatorID(8 byte)	随机数
RootKeyIndex(1 byte)	0x00
SessionKey(1 byte)	0x01
EncAlg(2 byte)	0x0100
MDInfo(5 byte)	随机数
Reserved(6 byte)	0
CheckSum(2 byte)	对消息头和消息体数据进行CheckSum校验

无效的连接请求 CONNECT\_REQ(e)的 'CheckSum' 值与正确值不同, 其余内容与上表一致。

## A.4 连接响应

有效的连接响应 CONNECT\_RSP(t)的定义如下:

字段	值
Rfu (4 bit)	0
FormatType(4 bit)	8
Status(8 bit)	0x00
MsgCode(8 bit)	18
MsgLen(16 bit)	24
Result(1 Byte)	0x00
RootKeyIndex(1 Byte)	0x00
SessionKey(1 Byte)	0x01
EncAlg(2 Byte)	0x0100
SDInfo(5 Byte)	自定义
SDRand (8 Byte)	随机数
Reseved(6 Byte)	0
CheckSum(2 Byte)	对消息头和消息体数据进行CheckSum校验

## A.5 数据交换请求

数据交换请求包括 APDATA\_REQ(t1), APDATA\_REQ(t2), APDATA\_REQ(t3), APDATA\_REQ(t4) 和 APDATA\_REQ(e)

有效的数据交换请求 APDATA\_REQ(t1)定义如下表所示:

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0
MsgCode(8bit)	19
MsgLen(16 bit)	24
EncPayLoad	对明文数据 (0x99 0x99 0x00 0x00 + 0x0D + 13Byte随机数) 加密后得到的密文数据, 加密算法由连接阶段双方商定
Checksum(2 byte)	对消息头和消息体数据进行Checksum校验

错误的交换请求 APDATA\_REQ(e)的 'Checksum' 值与正确值不同, 其余内容与上表一致。

APDATA\_REQ(t2)定义如下表所示:

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0
MsgCode(8bit)	19
MsgLen(16 bit)	248
EncPayLoad	对明文数据 (0x99 0x99 0x00 0x00 + 0xEE + 238Byte随机数) 加密后得到的密文数据, 加密算法由连接阶段双方商定
Checksum(2 byte)	对消息头和消息体数据进行Checksum校验

APDATA\_REQ(t3)为消息长度超出 YD/T XXX.10 的数据交换请求指令, 定义如下表所示:

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0
MsgCode(8bit)	19
MsgLen(16 bit)	296
EncPayLoad	对明文数据 (x99 0x99 0x00 0x00 + 0x01 0x22 + 290Byte随机数) 加密后得到的密文数据, 加密算法由连接阶段双方商定
Checksum(2 byte)	对消息头和消息体数据进行Checksum校验

APDATA\_REQ(t4)为有效数据请求指令, 其定义与 APDATA\_REQ(t1)相同, 但响应方 DUT 对 APDATA\_REQ(t4)指令的处理时间超过 500ms。



## A.6 数据交换响应

有效的数据响应 APDATA\_RSP(t)与当前数据交换请求命令 APDATA\_REQ 对应。

若当前数据交换请求命令为 APDATA\_REQ(t1), 则 APDATA\_RSP(t)定义如下:

错误的 APDATA\_RSP(e), 'Checksum' 值与正确值不相同。

字段	值
Rfu (4 bit)	0
FormatType(4 bit)	8
Status(8 bit)	0
MsgCode(8 bit)	20
MsgLen(16 bit)	24
EncPayload(24 Byte)	对当前APDATA_REQ(t1)的数据内容进行解密得到明文数据, 将明文数据 '0x99 0x99 0x00 0x00' 去掉, 并在其尾部加上 '0x90 0x00', 然后将处理后的明文数据再加密
Checksum(2 Byte)	对消息头和消息体数据进行Checksum校验

若当前数据交换请求命令为 APDATA\_REQ(t2), 则 APDATA\_RSP(t)定义如下:

错误的 APDATA\_RSP(e), 'Checksum' 值与正确值不相同。

字段	值
Rfu (4 bit)	0
FormatType(4 bit)	8
Status(8 bit)	0
MsgCode(8 bit)	20
MsgLen(16 bit)	248
EncPayload(248 Byte)	对当前APDATA_REQ(t2)的数据内容进行解密得到明文数据, 将明文数据 '99 99 00 00' 去掉, 并在其尾部加上 '90 00', 然后将处理后的明文数据再加密
Checksum(2 Byte)	对消息头和消息体数据进行Checksum校验

## A.7 长时等待命令

有效的长时等待命令 LTW(t)定义如下:

字段	值
Rfu (4 bit)	0
FormatType(4 bit)	8
Status(8 bit)	0
MsgCode(8 bit)	25
MsgLen(16 bit)	2
RandData(1Byte)	随机数
Reserved(1 Byte)	0
Checksum(2 Byte)	对消息头和消息体数据进行Checksum校验



无效的长时等待命令 LTW(e) 的 ‘Checksum’ 值与正确值不同，其余内容与上表一致。

#### A.8 维持连接请求

有效的维持连接请求 LINKCTL\_REQ(t)定义如下：

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0
MsgCode(8bit)	22
MsgLen(16 bit)	2
RandData(1 byte)	随机数
Reserved(1 byte)	0
Checksum(2 byte)	对消息头和消息体数据进行Checksum校验

无效的维持连接请求 LINKCTL\_REQ(e) 的 ‘Checksum’ 值与正确值不同，其余内容与上表一致。

#### A.9 维持连接响应

有效的维持连接请求 LINKCTL\_RSP(t)定义如下：

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0
MsgCode(8bit)	23
MsgLen(16 bit)	2
RandData (1 byte)	随机
Reserved (1 byte)	0
Checksum (2 byte)	对消息头和消息体数据进行Checksum校验

无效的维持连接请求 LINKCTL\_RSP(e) 的 ‘Checksum’ 值与正确值不同，其余内容与上表一致。

#### A.10 冲突检测命令

有效的冲突检测命令 CHECK1\_REQ(t)定义如下：

字段	值
MsgCode(4bit)	2
MsgLen(4 bit)	2
CDC(2 byte)	取当前响应方生成的随机数IDs的前2字节

无效的冲突检测命令 CHECK1\_REQ(e) 的 ‘CDC’ 值与正确值不同，其余内容与上表一致。

## A.11 冲突响应消息

冲突响应消息 CHECK\_RSP(t1)定义如下:

属性	数值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0 or 1
MsgCode(8bit)	24
MsgLen(16 bit)	16
RandData (1 byte)	随机
TargetID(8 byte)	响应方标识
Reserved (7 byte)	0
Checksum (2 byte)	对消息头和消息体数据进行Checksum校验

冲突响应消息 CHECK\_RSP(t2) 的 ‘TargetID’ 值与当前接入的卡片 ID 不同, 其余内容与上表一致。

## A.11 连接确认

有效的连接确认消息 CHECK2\_REQ(t)定义如下:

字段	值
MsgCode(4bit)	3
MsgLen(4bit)	2
TRI(2 Byte)	取当前响应方生成的随机数IDs的前2字节

无效的连接确认消息 CHECK2\_REQ(e) 的 ‘CDC’ 值与正确值不同, 其余内容与上表一致。

## A.12 关闭连接请求

有效的关闭连接请求 CLOSE\_REQ(t1)定义如下:

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0
MsgCode(8bit)	26
MsgLen(16 bit)	4
NeedResp(1 byte)	1
Reserved(3 byte)	0
Checksum(2 byte)	对消息头和消息体数据进行Checksum校验

有效的关闭连接请求 CLOSE\_REQ(t2)的 ‘NeedResp’ 值为 0, 其余内容与上表一致。

无效的关闭连接请求 CLOSE\_REQ(e) 的 ‘Checksum’ 值与正确值不同, 其余内容与上表一致。

## A.13 关闭连接响应

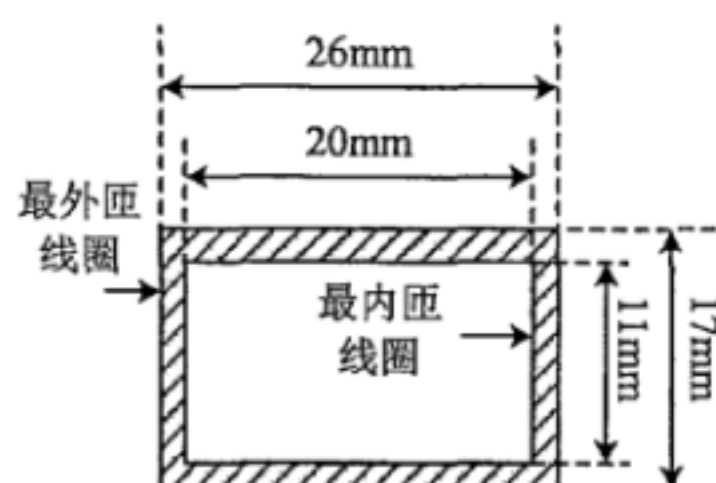
有效的关闭连接响应 CLOSE\_RSP (t)定义如下:

字段	值
Rfu(4 bit)	0
FormatType(4bit)	8
Status(8bit)	0
MsgCode(8bit)	27
MsgLen(16 bit)	4
CloseResult(1 byte)	0
Reserved(3 byte)	0
Checksum(2 byte)	对消息头和消息体数据进行Checksum校验

附录 B  
(规范性附录)  
标准感应线圈

标准感应线圈为在PCB上制作而成的印制线圈，在第一层上绕制9匝线圈后，通过过孔连接至下一层再绕制9匝线圈，以此类推，共完成6层线圈的绕制，6层线圈的绕制方向保持一致(同为顺时针方向或同为逆时针方向)，共54匝线圈。线圈的线宽均为0.15mm，线间距均为0.15mm，其单层结构如图B.1所示。标准感应线圈其他主要参数如下。

- 介质材料：FR4；
- 介质厚度：0.32mm；
- 布线层数：6；
- 导体材料：铜；
- 导体厚度：35 $\mu$ m；
- 成品厚度：1.6mm  $\pm$ 10%；
- 直流电阻：12.5ohm  $\pm$ 15%。

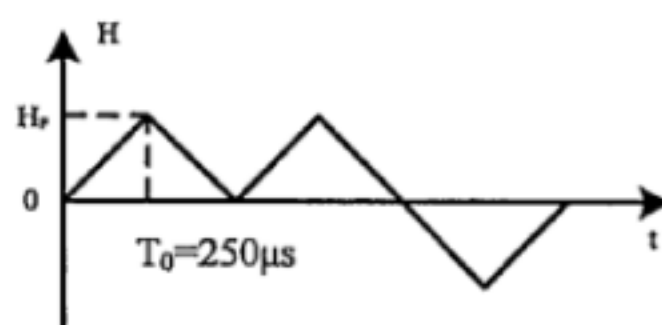


图B.1 标准感应线圈单层结构示意图



附录 C  
(资料性附录)

磁通道感应电压与场强的推导



图C.1 标准磁通道磁场信号调制波形

标准的磁通道磁场信号调制波形如图C.1所示,它是等斜率(绝对值)的,所以标准感应线圈处于该磁场中时,感应到的电压峰值应为:

$$V_p = \frac{N \times d\phi}{dt} = \frac{N \times d(B \times S)}{dt}$$

因为在标准感应线圈范围内的磁场强度呈较均匀的分布,所以面积 $S$ 范围内, $B$ 恒定,所以上式可以转换为:

$$V_p = \frac{N \times d(B \times S)}{dt} = \frac{N \times S \times dB}{dt} = \frac{\mu_0 \times N \times S \times dH}{dt}$$

由图 C.1 可知,  $\frac{dH}{dt} = \frac{H_p}{T_0}$

所以,  $V_p = \mu_0 NS \frac{H_p}{T_0}$

所以,  $H_p = \frac{V_p T_0}{\mu_0 NS} = \frac{V_{PP} T_0}{2\mu_0 NS} = KV_{PP}$

式中  $K = \frac{T_0}{2\mu_0 NS}$

上述推导中:

- $H_p$  为磁场强度峰值;
- $V_p$  为感应电压峰值;
- $V_{PP}$  为感应电压峰峰值;
- $T_0$  为符号周期(250us);
- $N$  为标准感应线圈匝数;
- $S$  为标准感应线圈等效面积;
- $\mu_0$  为真空磁导率( $4\pi \times 10^{-7}$ )。

中华人民共和国  
通信行业标准

手机支付

基于 2.45GHz RCC(限域通信)技术的非接触射频接口测试方法

YD/T 2773-2014

\*

人民邮电出版社出版发行

北京市丰台区成寿寺路 11 号邮电出版大厦

邮政编码: 100164

北京康利胶印厂印刷

版权所有 不得翻印

\*

开本: 880×1230 1/16

2015 年 12 月第 1 版

印张: 5

2015 年 12 月北京第 1 次印刷

字数: 139 千字

15115 • 604

定价: 50 元

本书如有印装质量问题, 请与本社联系 电话: (010)81055492