【**系统管理平台**】

**接口设计说明书**

【SMP-IDD-V0.1】

拟 制

审 核

标准化

批 准

深圳市中航比特通讯技术有限公司

**修订历史记录**

**变更类型 A** - 增加 **M** - 修订 **D** - 删除

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 变更  版本号 | 日期 | 变更  类型 | 修改人 | 摘要 | 备注 |
| V0.1 | 2013.07.02 | A | 曹文科 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1 范围 5

1.1 标识 5

1.2 系统概述 5

1.3 文档概述 5

2 引用文档 6

3 接口设计 6

3.1 接口标识和接口图 6

3.2 I-IF1： 6

3.2.1 应用管理 6

3.2.2 客户端XmlSql操作信令 7

3.2.3 应用数据上报 8

3.3 I-IF2： 8

3.3.1 数据动态同步信令XmlDbSync (S-E-PIP & S-I-PIP) 8

3.3.2 数据传输信令XmlDbTransfer(S-I-PIP | S-E-PIP) 8

3.3.3 数据前转信令ForwardXmlSql(S-E-PIP) 9

3.4 I-IF3： 10

3.5 I-IF4： 10

3.5.1 数据存储总体方案 10

3.5.2 <Table> … </Table> 11

3.5.3 <Record> … </Record > 11

3.5.4 <Field/> 12

3.5.5 内建数据字典（~O~鸡与蛋的故事~O~） 12

3.5.6 操作日志 13

3.6 E-IF1： 13

3.4.1 客户端注册(G-VIP & S-E-VIP) 13

3.4.2 权限访问信令(S-E-VIP) 14

3.4.3 XML-SQL操作信令(S-E-VIP) 15

3.6.4 自发现管理 15

3.7 E-IF2： 16

3.7.1 客户端XmlSql操作信令 16

3.7.2 应用数据上报 16

4 模型设计 17

4.1 数据模型 17

4.1.1 表格化数据 17

4.1.2 XML模型抉择 18

4.1.3 表格模型描述 18

4.1.4 系统目录的规划 22

4.1.5 统一编码 23

4.1.6 XML-SQL 24

4.2 通信模型 27

4.2.1 组网模型 28

4.2.2 可靠通信 30

4.3 架构建议 45

4.3.1 系统管理服务器架构设计建议 45

4.3.2 单板系统管理模块架构设计建议 46

4.3.3 业务系统数据管理部件架构设计建议 46

4.3.4 客户端数据管理部件架构设计建议 47

5 需求的可追踪性 47

5.1 部署需求 47

5.2 数据配置功能需求 48

5.3 故障管理功能需求 48

5.4 业务管理功能需求 49

5.5 系统监控功能需求 49

5.6 系统信息上报功能需求 49

5.7 软件管理功能需求 49

5.8 系统安全功能需求 50

6 注释 50

# 范围

## 标识

适用系统：SMP：System Manage Platfram 系统管理平台 版本：V0.1

接口实体：

SMS：System Manage Server 系统管理服务器 版本V0.1

SMU：uni-board System Manage Unit，单板系统管理模块 版本V0.1

SMC：System Manage Client 系统管理客户端 版本V0.1

DBS：Database System数据库系统

FS：File System文件系统

AS：Application Server应用服务器

接口标识：

I-IF1：SMU与SMS之间的接口

I-IF2：SMS与SMS之间的接口

E-IF1：SMC与SMS之间的接口

E-IF2：AS与SMU之间的接口

I-IF3：SMS与DBS之间的接口

I-IF4：SMS与FS之间的接口

适用时：软件设计、软件实现、单元测试、集成测试、系统测试、合格性测试。

## 系统概述

系统管理平台系公司独立项目，为模块化通信平台的组成部分，主要为模块化通讯平台相关产品提供管理与维护功能，以方便用户对产品进行相关的管理与维护操作。

系统管理平台的目标是为模块化通信平台产品提供通用化的系统管理功能支撑，对产品中系统管理的功能需求进行抽象化，使用通用的逻辑进行处理，保持系统管理业务逻辑上的一致，以屏蔽不同产品在数据上的差异性。要做到在产品的具体实施过程中，只需要对系统管理平台做一些必要的配置，即可实现并满足产品对系统管理方面的功能需求。

系统管理平台需要有支持灵活部署的能力，以适配不同产品的不同部署需求。要能够适应简化的部署方式，也要能够适应多层级的部署方式。

系统管理平台对外提供统一功能接口，包括系统管理对客户端接口以及对业务应用接口。

系统管理平台主要包括以下功能：数据配置功能、故障管理功能、业务管理功能、系统监控功能、系统信息上报功能、软件管理功能、系统安全功能等。

## 文档概述

本文档用于指导系统管理平台及其客户端的部分模块设计，也可供系统管理平台系统测试人员系统测试进行参考。

本文档集中描述系统管理平台与客户端和应用系统之间的通信接口。

# 引用文档

SMP\_SSS\_V1.0 《系统管理平台—系统规格说明书》

SMP\_SSDD\_V0.1《系统管理平台—系统设计说明书》

# 接口设计

## 接口标识和接口图



|  |  |
| --- | --- |
| 接口名 | 接口描述 |
| I-IF1 | 单板系统管理模块和系统管理服务器之间接口 |
| I-IF2 | 系统管理服务器之间的通信接口，用于数据同步和前转工作。 |
| E-IF1 | 系统管理客户端与系统管理平台之间的接口 |
| E-IF2 | 应用系统与系统管理平台之间的接口 |

本章所描述的各种接口都依赖于第4章的模型设计。

### 节点编码

4.2.2.5.2基本信令定义中需要填写‘发端节点编号’：

(1)客户端在发送时，用自身的IP地址的uint32形式表示。

(2)系统管理服务器、单板模块在发送消息时，用0x00XXYYZZ，其中XX表示MDN编号，YY表示MUN，ZZ表示SSN，先定义系统管理服务器的SSN为0，单板模块的SSN为1，其它由应用自身定义。

(3) 应用在发送消息时，用0x00FFFFZZ表示，其中ZZ表示应用自身的SSN。大于1。

当最高8位为非零时，表示接入为客户端。

### 扩展插件

扩展定义以下几个序列化插件：

|  |  |  |
| --- | --- | --- |
| 插件号 | 插件描述 | 接口 |
| 1 | SMP插件 |  |

## 信令汇总表（按管理划分，共55条业务信令）

### 客户端管理信令

|  |  |  |  |
| --- | --- | --- | --- |
| 名称(nOp) | 接口 | 方向 | 说明 |
| SmcLoginReq(0) | E-IF1 | SMC->SMS | SMC向SMS注册，客户端注册信息需要在管理范围之间同步。 |
| SmcLoginResp(1) | E-IF1 | SMS->SMS | SMS对SMC注册做出响应，连带把权限返回。 |
| SmcLogout(2) | E-IF1 | SMC->SMS | SMC向SMS注销，这是可选的，SMS应该可以根据链路状态判断SMC是否还在线。 |

### 应用配置管理信令

|  |  |  |  |
| --- | --- | --- | --- |
| 名称(nOp) | 接口 | 方向 | 说明 |
| AppCfgReq(3) | E-IF1 | SMC->SMS | SMU发起应用配置更新请求，SMS收到请求，如果是本管理范围的应用配置，立即做本地存储，否则通过AppCfgForwardReq前转到相应SMS。 |
| AppCfgResp(4) | E-IF1 | SMS->SMC | SMS做本地存储后，或收到归属SMS的响应AppCfgForwardResp后，需要回该响应。 |
| AppCfgForwardReq(5) | I-IF2 | SMS->SMS | 当SMS判断自己不是AppCfgReq所标示的归属SMS时，需要转发到相应的归属SMS，当收到AppCfgForwardReq时需要做本地存储。 |
| AppCfgForwardResp(6) | I-IF2 | SMS->SMS | 当对AppCfgForwardReq做好本地存储后返回AppCfgForwardResp。 |
| AppCfgDownReq(7) | I-IF1 | SMS->SMU | 归属SMS收到AppCfgReq或AppCfgForwardReq时，做好本地存储后，进行下发请求 |
| AppCfgDownResp(8) | I-IF1 | SMU->SMS | 单板收到AppCfgDownReq，做本地存储后，对系统管理服务器做回应。 |
| AppCfgNotify(9) | E-IF2 | SMU->APP | 单板收到AppCfgDownReq，并做本地存储后，需要对应用发起变更通知。 |
| AppCfgSync(10) | I-IF2 | SMS->SMS | 归属SMS收到AppCfgReq或AppCfgForwardReq时，做好本地存储后，即需要发起AppCfgSync以做管理范围内的配置同步。 |
| AppCfgQueryReq(11) | E-IF1 | SMC->SMS | SMC发起对应用配置查询的请求 |
| AppCfgQueryResp(12) | E-IF1 | SMS->SMC | SMS直接返回应用配置给客户端 |
| AppCfgQueryForwardReq(13) | I-IF2 | SMS->SMS | 当查询的内容不在当前管理范围内时 |
| AppCfgQueryForwardResp(14) | I-IF2 | SMS->SMS | 当查询客户端不在当前管理范围内时 |

### 平台数据管理信令

系统管理服务器的配置项：

（1）管理地址【IP和Port】；--不一定是配置，可用算法做计算，为方便调试，建议支持本地的配置方式。

（2）故障管理开关；

（3）历史告警容量阀值，事件表容量阀值。

（4）用户管理配置【有增删改】：

用户名、密码、权限。

权限定义：平台管理能力、应用管理能力、故障监控能力(可指定管理范围)

应用管理能力：管理能力主要是配置修改和业务控制。

平台管理能力：平台的配置修改和应用启停。

单板配置：

1. CPU\_UP、CPU\_Down、Memory\_Up、Memory\_Down
2. 应用管理配置策略配置，也只有修改

数据查询：

软件数据和状态数据需要到单板进行查询，其它数据直接在系统管理服务器查询即可（包括告警和事件数据，客户端用户管理配置）

|  |  |  |  |
| --- | --- | --- | --- |
| 名称(nOp) | 接口 | 方向 | 说明 |
| SmpCfgReq(15) | E-IF1 | SMC->SMS | SMU发起平台配置更新请求，SMS收到请求，如果是本管理范围的配置，立即做本地存储，否则通过SmpCfgForwardReq前转到相应SMS。 |
| SmpCfgResp(16) | E-IF1 | SMS->SMC | SMS做本地存储后，或收到归属SMS的响应SmpCfgForwardResp后，需要回该响应。 |
| SmpCfgForwardReq(17) | I-IF2 | SMS->SMS | 当SMS判断自己不是SmpCfgReq所标示的归属SMS时，需要转发到相应的归属SMS，当收到SmpCfgForwardReq时需要做本地存储。 |
| SmpCfgForwardResp(18) | I-IF2 | SMS->SMS | 当对SmpCfgForwardReq做好本地存储后返回SmpCfgForwardResp。 |
| SmpCfgDownReq(19) | I-IF1 | SMS->SMU | 若是单本的配置，SMS需要则发起该消息。 |
| SmpCfgDownResp(20) | I-IF1 | SMU->SMS | 单板对修改做回应 |
| SmpSync(21) | I-IF2 | SMS->SMS | 1. 归属SMS收到SmpCfgReq或SmpCfgDownReq时，做好本地存储后，即需要发起SmpSync以做配置同步。 2. 用户管理信息、客户端登陆信息、软件列表信息，需要在全局进行同步。 3. 软件管理策略信息、单板监控配置信息，需要在管理范围内同步。 |
| SmpQueryReq(22) | E-IF1 | SMC->SMS | SMC发起对平台数据查询的请求 |
| SmpQueryResp(23) | E-IF1 | SMS->SMC | SMS直接返回应用配置给客户端 |
| SmpQueryForwardReq(24) | I-IF2 | SMS->SMS | 当查询的内容不在当前管理范围内时 |
| SmpQueryForwardResp(25) | I-IF2 | SMS->SMS | 当查询客户端不在当前管理范围内时 |
| SmpQueryDownReq(26) | I-IF1 | SMS->SMU | 对SMU监控信息、软件运行状态信息的查询，SMS需要下发到相应SMU。 |
| SmpQueryDownResp(27) | I-IF1 | SMU->SMS | SMU对查询做回应 |

### 告警事件管理信令

|  |  |  |  |
| --- | --- | --- | --- |
| 名称(nOp) | 接口 | 方向 | 说明 |
| ReportAlarm(28) | E-IF2  I-IF1  I-IF2  E-IF1 | APP->SMU  SMU->SMS  SMS->SMS  SMS->SMC | 告警报告，  SMU转发或自发告警  告警同步  告警下发 |
| ReportEvent(29) | E-IF2  I-IF1  I-IF2  E-IF1 | APP->SMU  SMU->SMS  SMS->SMS  SMS->SMC | 事件报告，  SMU转发或自发事件  事件同步  事件下发 |

### 业务控制信令

|  |  |  |  |
| --- | --- | --- | --- |
| 名称(nOp) | 接口 | 方向 | 说明 |
| SvrCtrlReq(30) | E-IF1 | SMC->SMS | SMU发起业务控制请求 |
| SvrCtrlResp(31) | E-IF1 | SMS->SMC | SMS对控制做出回应。 |
| SvrCtrlForwardReq(32) | I-IF2 | SMS->SMS | 当SMS判断自己不是SvrCtrlReq所标示的归属SMS时，需要转发到相应的归属SMS |
| SvrCtrlForwardResp(33) | I-IF2 | SMS->SMS | 当收到SvrCtrlDownResp时，如果该会话是通过转发而来的，那么需要发起SvrCtrlForwardResp |
| SvrCtrlDownReq(34) | I-IF1 | SMS->SMU | 归属SMS收到SvrCtrlReq或SvrCtrlForwardReq时，进行下发请求 |
| SvrCtrlDownResp(35) | I-IF1 | SMU->SMS | SMU收到SvrCtrlDownReq，下发SvrCtrlNotifyReq，并受到SvrCtrlNotifyResp后才回应SvrCtrlDownResp。 |
| SvrCtrlNotifyReq(36) | E-IF2 | SMU->APP | SMU收到SvrCtrlDownReq后，需要发起控制通知。 |
| SvrCtrlNotifyResp(37) | E-IF2 | APP->SMU | 应用控制通知响应。 |

注意SMS和SMU均不介入业务控制的具体事务中，仅仅充当传输通道，是否有回应，可完全有APP做决定，SMC和APP一致即可。

将来由SMC发起直接操作APP的各种信令都可以在这个信令基础上扩展。

### 软件管理信令

#### 软件启停控制信令

|  |  |  |  |
| --- | --- | --- | --- |
| 名称(nOp) | 接口 | 方向 | 说明 |
| SwCtrlReq(38) | E-IF1  I-IF1  I-IF2 | SMC->SMS  SMS->SMU  SMS->SMS | 1. SMU发起应用应用启停控制请求，无需响应，可通过设计运行状态查看。 2. 仅需要到SMU即可，I-IF2用于前转。 |

#### 软件升级的备份信令

|  |  |  |  |
| --- | --- | --- | --- |
| 名称(nOp) | 接口 | 方向 | 说明 |
| SwBakReq(39) | E-IF1 | SMC->SMS | SMC发起应用软件下载请求 |
| SwBakResp(40) | E-IF1 | SMS->SMC | SMS收到SwBakDownResp或SwBakForwardResp时，应当做出该回应。 |
| SwBakForwardReq(41) | I-IF2 | SMS->SMS | 当SMS判断自己不是SwBakReq所标示的归属SMS时，需要转发到相应的归属SMS |
| SwBakForwardResp(42) | I-IF2 | SMS->SMS | 当收到SwBakDownResp时，如果该会话是通过转发而来的，那么需要发起SwBakForwardResp。 |
| SwBakDownReq(43) | I-IF1 | SMS->SMU | 归属SMS收到SwBakReq或SwBakForwardReq时，进行下发请求 |
| SwBakDownResp(44) | I-IF1 | SMU->SMS | SMU收到SwBakDownReq，返回软件包。 |

#### 软件升级信令

|  |  |  |  |
| --- | --- | --- | --- |
| 名称(nOp) | 接口 | 方向 | 说明 |
| SwUgdReq(45) | E-IF1 | SMC->SMS | SMC发起应用软件升级请求 |
| SwUgdResp(46) | E-IF1 | SMS->SMC | SMS收到SwUgdDownResp或SwUgdForwardResp时，应当做出该回应。。 |
| SwUgdForwardReq(47) | I-IF2 | SMS->SMS | 当SMS判断自己不是SwUgdReq所标示的归属SMS时，需要转发到相应的归属SMS |
| SwUgdForwardResp(48) | I-IF2 | SMS->SMS | 当收到SwUgdDownResp时，如果该会话是通过转发而来的，那么需要发起SwUgdForwardResp。 |
| SwUgdDownReq(49) | I-IF1 | SMS->SMU | 归属SMS收到SwUgdReq或SwUgdForwardReq时，进行下发请求。 |
| SwUgdDownResp(50) | I-IF1 | SMU->SMS | SMU收到SwUgdDownReq，做好本地存储后即可返回该响应。随后自行停止应用，并删除旧软件，SMC通过状态查询发现软件停止后，如果配置文件需要升级，可将升级后配置文件发给SMS，在通过软件启动控制命令进行升级启动。 |

软件升级的版本验证由客户端进行，在SMS中做一个内定的配置文件，记录软件运行许可信息，该文件由公司加密发布，由客户端通过文件上传信令【借用AppCfgReq】传SMS，SMS服务器解密存储。

SMC进行版本升级前，先下载对应的版本信息（不能存储），自行校验。软件版本信息约为：应用名，应用版本号，许可的硬件类型，许可的硬件版本。上述四个字段联合唯一。

如果软件升级需要配置文件格式升级，客户端需要先整合旧数据进新格式的配置文件，然后再开始启动备份、上传软件、监控程序状态停止后，上传新配置文件，执行软件启动控制。如果不需要升级配置文件，那么SMU收到SwUgdDownReq进行本地存储后，即可停止应用并重启。

#### 软件运行状态管理

|  |  |  |  |
| --- | --- | --- | --- |
| 名称(nOp) | 接口 | 方向 | 说明 |
| SwStatusReport(51) | E-IF2 | APP->SMU | 应用向SMU报告状态：   1. Starting;//此时不允许客户端更新相关配置，流程由SMS控制。 2. Started; //需要定时报告 3. Stoped; //退出时报告，柔性退出。 |

### 文件传输校验信令

当SMS启动时，向任一处于服务状态的SMS请求全局数据传输，并向管理范围内的服务状态的SMS请求该管理范围的配置数据传输。

当SMU启动时，会向归宿SMS请求该单板的配置数据传输。

|  |  |  |  |
| --- | --- | --- | --- |
| 名称(nOp) | 接口 | 方向 | 说明 |
| SmsTransferReq(52) | I-IF2 | SMS->SMS |  |
| SmsTransferResp(53) | I-IF2 | SMS->SMS |  |
| SmuTransferReq(54) | I-IF1 | SMU->SMS |  |
| SmuTransferResp(55) | I-IF1 | SMU->SMS |  |

### 令牌竞选信令

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 接口 | 方向 | 说明 |
| TokenBootStrap | I-IF2 | SMS->SMS | 参考4.2.2.5.4章节 |

## E-IF1 (SMC-SMS)：

### SmcLogin & SmcLogout

（1）SmcLoginReq消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| UserName | string | 用户名最长16个字符，传输时包含结尾’\0’字符 |
| AuthCode | uint8[16] | 由UserName[16]+ UserPsw[10]+Time(yyyymmddhhmmss)生成的MD5编码。UserName和UserPsw不足位填0 |
| Time | Uint32 | 登陆时间。 |

（2）SmcLoginResp消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| Status | uint | 0=成功；1=鉴权失败【不区分用户不存在还是密码不正确】。 |
| License | uint | 授权码；登陆成功后，会返回一个授权码。  该授权码在断链或登出之前一直有效**。** |
| Roles | string | 1. SMP:表示能对平台进行管理。 2. *APP*:表示能对应用进行管理，这里应用名，ALL表示对所有应用都能管理。 3. MON:MDNS(用-分隔的管理范围列表，255表示全监控)：指监控能力，可指定多个管理范围。 4. 用逗号分隔的一个字符串表示。 |

（3）SmcLogout

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| License | uint | 授权码 |

### AppCfg

1. AppCfgReq消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| License | uint32 | 授权码 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmcSession | uint32 | SMC的请求会话编号 |
| Dynamic | Uint8 | 0=静态配置，应用需要重启，客户端在发起该信令前最好显示的提醒客户，并让客户确认。  1=动态配置，应用不需要重启。 |
| XmlName | string | Xml文件名，由服务器决定该文件放在何处。 |
| XmlData | Uint8[] | Xml文件数据，其大小由包头Size控制。 |

（2）AppCfgResp消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SmcSession | uint32 | SMC的请求会话编号 |
| Result | uint32 | 0=成功；1=路由失败；2=存储失败 |

SMS写文件成功后，即回应成功，并启动下发单板流程，下发单板可能失败，就需要重新下发单板，如果三次下发单板失败，SMS将发告警。

（3）AppCfgQueryReq消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| License | uint32 | 授权码 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmcSession | uint32 | SMC的请求会话编号 |
| XmlName | string | Xml文件名，由服务器决定该文件放在何处。 |

（4）AppCfgQueryResp消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SmcSession | uint32 | SMC的请求会话编号 |
| Result | uint32 | 0=成功；1=路由失败；2=文件读取失败 |
| XmlData | Uint8[] | Xml文件数据，其大小由包头Size控制。 |

### SmpData

1. SmpCfgReq消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| License | uint32 | 授权码 |
| MDN | Uint8 | 目标MDN |
| MUN | Uint8 | 单板号，255表示SMS，否则表示具体的单板配置 |
| SmcSession | uint32 | SMC的请求会话编号 |
| Dynamic | Uint8 | 0=静态配置，应用需要重启，客户端在发起该信令前最好显示的提醒客户，并让客户确认。  1=动态配置，应用不需要重启。 |
| TableName | string | 配置表名 |
| Operator | Uint32 | 0=Insert;1=Update;2=Delete |
| Data | Uint8[] | （1）FieldName=FieldValue的文本列表，用’\0’字符分隔，用双’\0’结束。  （2）Update操作：索引字段不可修改，若要修改索引字段，采用先删后插的两条信令解决，SMS根据索引字段，以修改其它字段的值。 |

（2）SmpCfgResp消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SmcSession | uint32 | SMC的请求会话编号 |
| Result | uint32 | 0=成功；1=无效配置（可能表名不存在，可能字段名不存在等） |

SMS写文件成功后，即回应成功，并启动下发单板流程，下发单板可能失败，就需要重新下发单板，如果三次下发单板失败，SMS将发告警。

（3）SmpQueryReq消息体

|  |  |  |
| --- | --- | --- |
| License | uint32 | 授权码 |
| MDN | Uint8 | 目标MDN |
| MUN | Uint8 | 单板号，255表示SMS，否则表示具体的单板配置 |
| SmcSession | uint32 | SMC的请求会话编号 |
| TableName | string | 配置表名，仅支持整表查询 |

（4）SmpQueryResp消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SmcSession | uint32 | SMC的请求会话编号 |
| Result | uint32 | 0=成功；1=路由失败；2=文件读取失败 |
| Data | Uint8[] | 是多行数据集，行之间用双’\0’分隔，行内字段之间以单’\0’分隔，用四个’\0’作为整个数据的结束。第一行是字段名列表，其它行为记录的文本数据。 |

### Alarm & Event

用于告警和事件监控。

1. ReportAlarm

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| AlarmClass | Uint8 | 告警类别 |
| AlarmType | Uint8 | 告警类型: 1=告警,2=恢复,3=复位 |
| AlarmTime | Uint32 | 告警时间: 告警发生时间,单位:秒 |
| AlarmObjType | Uint8 | 告警对象类型 |
| AlarmObjId | Uint8 | 告警对象标识 |
| AlarmInfoLen | Uint16 | 告警信息长度，最长200字节 |
| AlarmInfo | Uint8[] | 告警信息，长度为AlarmInfoLen |

1. ReportEvent

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| EventClass | Uint8 | 事件类别 |
| EventTime | Uint32 | 事件发生时间 |
| EventObjType | Uint8 | 事件对象类型 |
| EventObjId | Uint8 | 事件对象标识 |
| EventInfoLen | Uint16 | 告警信息长度，最长200字节 |
| EventInfo | Uint8[] | 告警信息，长度为AlarmInfoLen |

如果SMS不管理任何故障，则不需要剔重过滤，否则需要。

### SvrCtrl

（1）SvrCtrlReq

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| License | uint32 | 授权码 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmcSession | uint32 | SMC的请求会话编号。 |
| CtrlReqData | Uint8[] | 控制请求数据，其大小由包头Size控制。 |

（2）SvrCtrlResp

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SmcSession | uint32 | SMC的请求会话编号。 |
| Result | Uint8[] | 控制响应数据，其大小由包头Size控制。 |

### SwManage

（1）SwCtrlReq

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| License | uint32 | 授权码 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| Cmd | Uint32 | 0=启动；1=停止 |

（2）SwBakReq

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| License | uint32 | 授权码 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmcSession | uint32 | SMC的请求会话编号。 |

（3）SwBakResp

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SmcSession | uint32 | SMC的请求会话编号。 |
| Result | Uint32 | 0=成功；1=不存在；2=其它失败 |
| SwVersion | Uint32 | 软件版本； |
| SwData | Uint8[] | 软件文件数据，其大小由包头Size控制。  建议做成一个.tar.gz包(文件名包含版本号)，板上保留该包？ |

（4）SwUgdReq

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| License | uint32 | 授权码 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmcSession | uint32 | SMC的请求会话编号。 |
| SwVersion | Uint32 | 软件版本； |
| SwData | Uint8[] | 软件文件数据，其大小由包头Size控制。  建议做成一个.tar.gz包(文件名包含版本号)，板上保留该包？ |

（5）SwUgdResp

|  |  |  |
| --- | --- | --- |
| SmcSession | uint32 | SMC的请求会话编号。 |
| Result | Uint32 | 0=成功；1=失败，并不是升级失败，而是网络或写文件失败等。升级是否成功，需要看软件的运行状态。 |

## I-IF2 (SMS-SMS) ：

### 令牌竞选与VIP

参考4.2.5.5.4

### AppCfg

（1）AppCfgForwardReq消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmcSession | uint32 | SMC的请求会话编号 |
| Dynamic | Uint8 | 0=静态配置，应用需要重启。1=动态配置，应用不需要重启。 |
| XmlName | string | Xml文件名，由服务器决定该文件放在何处。 |
| XmlData | Uint8[] | Xml文件数据，其大小由包头Size控制。 |

（2）AppCfgForwardResp消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| SmcSession | uint32 | SMC的请求会话编号 |
| Result | uint32 | 0=成功；1=路由失败；2=存储失败 |

（3）AppCfgQueryForwardReq消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmcSession | uint32 | SMC的请求会话编号 |
| XmlName | string | Xml文件名，由服务器决定该文件放在何处。 |

（4）AppCfgQueryForwardResp消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| SmcSession | uint32 | SMC的请求会话编号 |
| Result | uint32 | 0=成功；1=路由失败；2=文件读取失败 |
| XmlData | Uint8[] | Xml文件数据，其大小由包头Size控制。 |

（5）AppCfgSync

在系统管理范围之内进行应用配置同步。

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| XmlName | string | Xml文件名，由服务器决定该文件放在何处。 |
| XmlData | Uint8[] | Xml文件数据，其大小由包头Size控制。 |

当SMS收到AppCfgReq或AppCfgForwardReq，并做好存储后，即可发起该流程。

### SmpData

1. SmpCfgForwardReq消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| MUN | Uint8 | 单板号，255表示SMS，否则表示具体的单板配置 |
| SmcSession | uint32 | SMC的请求会话编号 |
| Dynamic | Uint8 | 0=静态配置，应用需要重启。  1=动态配置，应用不需要重启。 |
| TableName | string | 配置表名 |
| Operator | Uint32 | 0=Insert;1=Update;2=Delete |
| Data | Uint8[] | （1）FieldName=FieldValue的文本列表，用’\0’字符分隔，用双’\0’结束。  （2）Update操作：索引字段不可修改，若要修改索引字段，采用先删后插的两条信令解决，SMS根据索引字段，以修改其它字段的值。 |

（2）SmpCfgForwardResp消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| SmcSession | uint32 | SMC的请求会话编号 |
| Result | uint32 | 0=成功；1=无效配置（可能表名不存在，可能字段名不存在等） |

（3）SmpQueryForwardReq消息体

|  |  |  |
| --- | --- | --- |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| MUN | Uint8 | 单板号，255表示SMS，否则表示具体的单板配置 |
| SmcSession | uint32 | SMC的请求会话编号 |
| TableName | string | 配置表名，仅支持整表查询 |

（4）SmpQueryForwardResp消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| MUN | Uint8 | 单板号，255表示SMS，否则表示具体的单板配置 |
| SmcSession | uint32 | SMC的请求会话编号 |
| Result | uint32 | 0=成功；1=路由失败；2=文件读取失败 |
| Data | Uint8[] | 是多行数据集，行之间用双’\0’分隔，行内字段之间以单’\0’分隔，用四个’\0’作为整个数据的结束。第一行是字段名列表，其它行为记录的文本数据。 |

（5）SmpSync

(a)软件列表是全局的，以便SMS根据SSN+InstanceID查找相应的MDN和MUN。

(b)用户管理信息、客户端登陆信息是全局的。

(b)软件策略，监控配置是管理范围内的。

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| MUN | Uint8 | 单板号，255表示全局范围，否则管理范围内的某单板配置 |
| TableName | string | 配置表名 |
| Operator | Uint32 | 0=Insert;1=Update;2=Delete |
| Data | Uint8[] | （1）FieldName=FieldValue的文本列表，用’\0’字符分隔，用双’\0’结束。  （2）Update操作：索引字段不可修改，若要修改索引字段，采用先删后插的两条信令解决，SMS根据索引字段，以修改其它字段的值。 |

### Alarm & Event

参考3.3.4，实际上是转发给客户端，如果SMS不管理任何故障，则不需要剔重过滤，否则需要。

### SvrCtrl

1. SvrCtrlForwardReq

|  |  |  |
| --- | --- | --- |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmcSession | uint32 | SMC的请求会话编号。 |
| CtrlReqData | Uint8[] | 控制请求数据，其大小由包头Size控制。 |

1. SvrCtrlForwardResp

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| SmcSession | uint32 | SMC的请求会话编号。 |
| Result | Uint8[] | 控制响应数据，其大小由包头Size控制。 |

### SwManage

（1）SwCtrlReq

请参考3.3.6.(1)

（2）SwBakForwardReq

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmcSession | uint32 | SMC的请求会话编号。 |

（2）SwBakForwardResp

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| SmcSession | uint32 | SMC的请求会话编号。 |
| Result | Uint32 | 0=成功；1=不存在；2=其它失败 |
| SwVersion | Uint32 | 软件版本； |
| SwData | Uint8[] | 软件文件数据，其大小由包头Size控制。  建议做成一个.tar.gz包(文件名包含版本号)，板上保留该包？ |

（4）SwUgdForwardReq

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmcSession | uint32 | SMC的请求会话编号。 |
| SwVersion | Uint32 | 软件版本； |
| SwData | Uint8[] | 软件文件数据，其大小由包头Size控制。  建议做成一个.tar.gz包(文件名包含版本号)，板上保留该包？ |

（5）SwUgdForwardResp

|  |  |  |
| --- | --- | --- |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| SmcSession | uint32 | SMC的请求会话编号。 |
| Result | Uint32 | 0=成功；1=失败，并不是升级失败，而是网络或写文件失败等。升级是否成功，需要看软件的运行状态。 |

### SmsTransfer

1. SmsTransferReq

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | | 数据类型 | 参数说明 | |
| Count | | Uint32 | 文件数 | |
|  | FileName | string | 可以是相对路径名 | 共Count项 |
| MD5 | Uint8[16] | MD5码 |

1. SmsTransferResp

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| Count | Uint32 | 文件总数，每次仅传一个文件，分多次传递。 |
| FileName | string | 可以是相对路径名 |
| MD5 | Uint8[16] | MD5码 |
| Flag | Uint8 | 0=更新或新建；1=删除 |
| FileData | Uint8[] | 由包头大小控制 |

需要先在跨管理范围间选择一个处于服务状态的SMS获取最新的全局数据，再在管理范围内的处于服务状态的SMS获取本管理范围内的局部数据。

## I-IF1 (SMU-SMS)：

### SmuTransfer

1. SmuTransferReq

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | | 数据类型 | 参数说明 | |
| MUN | | Uin8 | 单板编号 | |
| Count | | Uint32 | 文件数 | |
|  | FileName | string | 可以是相对路径名 | 共Count项 |
| MD5 | Uint8[16] | MD5码 |

1. SmuTransferResp

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| Count | Uint32 | 文件总数，每次仅传一个文件，分多次传递。 |
| FileName | string | 可以是相对路径名 |
| MD5 | Uint8[16] | MD5码 |
| Flag | Uint8 | 0=更新或新建；1=删除 |
| FileData | Uint8[] | 由包头大小控制 |

### AppCfg

1. AppCfgDownReq消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmsSession | uint32 | SMS的请求会话编号。  SMS需要做会话管理，SMS收到AppCfgReq后，本地保存后即可回应，该流程需要SMS单独发起（根据SSN和InstanceId算出需要发给那个单板）。 |
| Dynamic | Uint8 | 0=静态配置，应用需要重启。1=动态配置，应用不需要重启。 |
| XmlName | string | Xml文件名，由服务器决定该文件放在何处。 |
| XmlData | Uint8[] | Xml文件数据，其大小由包头Size控制。 |

（2）AppCfgDownResp消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SmsSession | uint32 | SMS的请求会话编号 |
| Result | uint32 | 0=成功；1=系统正忙(表示系统正在启动过程中) ；2=存储失败 |

### SmpData

1. SmpCfgDownReq消息体(SMS需要做会话管理)

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SmsSession | uint32 | SMS的请求会话编号。  SMS需要做会话管理，SMS收到SmpCfgReq后，本地保存后即可回应，该流程需要SMS单独发起。 |
| Dynamic | Uint8 | 0=静态配置，应用需要重启。1=动态配置，应用不需要重启。 |
| TableName | string | 配置表名 |
| Operator | Uint32 | 0=Insert;1=Update;2=Delete |
| Data | Uint8[] | （1）FieldName=FieldValue的文本列表，用’\0’字符分隔，用双’\0’结束。  （2）Update操作：索引字段不可修改，若要修改索引字段，采用先删后插的两条信令解决，SMS根据索引字段，以修改其它字段的值。 |

（2）SmpCfgDownResp消息体

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SmsSession | uint32 | SMS的请求会话编号 |
| Result | uint32 | 0=成功；1=系统正忙(表示系统正在启动过程中) ；2=存储失败 |

（3）SmpQueryDownReq

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| MDN | Uint8 | 是SMC投递的MDN，以方便信息返回 |
| SmcSession | uint32 | SMC的请求会话编号 |
| TableName | string | 配置表名，仅支持整表查询 |

（4）SmpQueryDownResp

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| MDN | Uint8 | 是SMC投递的MDN，以方便信息返回。  SMS收到SmpQueryDownResp，如果MDN不是当前MDN，需要发起SmpQueryForwardResp，否则直接发SmpQueryResp |
| SmcSession | uint32 | SMC的请求会话编号。 |
| Result | uint32 | 0=成功；1=查询失败 |
| Data | Uint8[] | 是多行数据集，行之间用双’\0’分隔，行内字段之间以单’\0’分隔，用四个’\0’作为整个数据的结束。第一行是字段名列表，其它行为记录的文本数据。 |

### Alarm & Event

参考3.3.4，实际上是转发给客户端，如果SMS不管理任何故障，则不需要剔重过滤，否则需要。

### SvrCtrl

SMS不需要做会话管理

（1）SvrCtrlDownReq

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| MDN | Uint8 | 是SMC投递的MDN，以方便信息返回 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmcSession | uint32 | SMC的请求会话编号。 |
| CtrlReqData | Uint8[] | 控制请求数据，其大小由包头Size控制。 |

（1）SvrCtrlDownResp

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| MDN | Uint8 | 是SMC投递的MDN，以方便信息返回  SMS收到SvrCtrlDownResp，如果MDN不是当前MDN，需要发起SvrCtrlForwardResp，否则直接发SvrCtrlResp |
| SmcSession | uint32 | SMC的请求会话编号。 |
| Result | Uint8[] | 控制响应数据，其大小由包头Size控制。 |

### SwManage

SMS不需要做会话管理

1. SwCtrlReq

参看3.3.6.(1)

（2）SwBakDownReq

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| MDN | Uint8 | 是SMC投递的MDN，以方便信息返回 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmcSession | uint32 | SMC的请求会话编号。 |

（3）SwBakDownResp

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| MDN | Uint8 | 是SMC投递的MDN，以方便信息返回  SMS收到SwBakDownResp，如果MDN不是当前MDN，需要发起SwBakForwardResp，否则直接发SwBakResp |
| SmcSession | uint32 | SMC的请求会话编号。 |
| Result | Uint32 | 0=成功；1=不存在；2=其它失败 |
| SwVersion | Uint32 | 软件版本； |
| SwData | Uint8[] | 软件文件数据，其大小由包头Size控制。  建议做成一个.tar.gz包(文件名包含版本号)，板上保留该包？ |

（4）SwUgdDownReq

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| MDN | Uint8 | 是SMC投递的MDN，以方便信息返回 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| SmcSession | uint32 | SMC的请求会话编号。 |
| SwVersion | Uint32 | 软件版本； |
| SwData | Uint8[] | 软件文件数据，其大小由包头Size控制。  建议做成一个.tar.gz包(文件名包含版本号)，板上保留该包？ |

（5）SwUgdDownResp

|  |  |  |
| --- | --- | --- |
| SMC | Uint32 | 客户端ID，实际上就是IP地址 |
| MDN | Uint8 | 是SMC投递的MDN，以方便信息返回  SMS收到SwUgdDownResp，如果MDN不是当前MDN，需要发起SwUgdForwardResp，否则直接发SwUgdResp |
| SmcSession | uint32 | SMC的请求会话编号。 |
| Result | Uint32 | 0=成功；1=失败，并不是升级失败，而是网络或写文件失败等。升级是否成功，需要看软件的运行状态。 |

## E-IF2：

### AppCfg

（1）AppCfgNotify

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| XmlName | string | Xml文件名，由服务器决定该文件放在何处。 |
| XmlData | Uint8[] | Xml文件数据，其大小由包头Size控制。 |

### Alarm & Event

参考3.3.4，应用发起告警和事件报告。

### SvrCtrl

SvrCtrlNotifyReq & SvrCtrlNotifyResp与SvrCtrlDownReq & SvrCtrlDownResp保持一致即可，单板不做会话管理。

### SwManage

（1）SwStatusReport

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SSN | Uint8 | 应用具体的类型编码(子系统号SubSystemNo) |
| InstanceId | Uint8 | 应用运行的实例ID，同一应用的不同实例的编号不能冲突。 |
| Status | Uint8 | (1)Starting;//此时不允许客户端更新相关配置，流程由SMS控制。  (2)Started; //需要定时报告，以防应用僵死。  (3)Stoped; //退出时报告，柔性退出。 |

# 模型设计

## 数据模型

### 配置数据

#### 应用配置数据

应用配置采用XML文件，其格式完全由应用自行定义，仅与系统管理客户端和应用自身相关，系统管理平台不参与其中，仅仅充当传输与存储的角色。

XML数据是树状结构，为了减小传输量，系统管理平台决定，以一颗树为单位进行文件存储于传输，业务应用系统可以含有多棵树，树的大小及深度由应用自行决定。

#### 平台配置数据

平台自身继续沿用传统的表格化配置数据，而不采用XML格式组织文件。

### 告警数据

#### 告警数据模型

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 | 数据源 |
| AlarmNo | Uint32 | 告警编号 | 平台 |
| AlarmClass | Uint8 | 告警类别 | 应用（大类） |
| AlarmType | Uint8 | 告警类型:  1=告警,2=恢复,3=复位 | 应用:1,2;  客户端:3，用于手工清除告警 |
| AlarmTime | Uint32 | 告警时间:  告警发生时间,单位:秒 | 应用（告警或自动恢复时间）,客户端（手工复位时间） |
| AlarmObjType | Uint8 | 告警对象类型 | 应用（小类） |
| AlarmObjId | Uint8 | 告警对象标识 | 应用 |
| AlarmLevel | Uint8 | 告警级别：  1=FATAL,2= CRITICAL,3= WARNING,4= INFO | 平台预配置 |
| AlarmInfoLen | Uint16 | 告警信息长度，最长200字节 | 应用 |
| AlarmInfo | Uint8[] | 告警信息，长度为AlarmInfoLen | 应用 |

#### 当前告警

AlarmNo, AlarmClass, AlarmTime,

AlarmObjType, AlarmObjId, AlarmLevel, AlarmInfoLen, AlarmInfo

由于AlarmType仅为1，所以该表格信息中不含该字段。

#### 历史告警

AlarmNo, AlarmClass, AlarmType, AlarmTime,

ClearTime,AlarmObjType, AlarmObjId, AlarmLevel, AlarmInfoLen, AlarmInfo

这里AlarmType取值为2或3，并发恢复时的AlarmTime填充到ClearTime。

#### 告警级别配置

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| AlarmClass | Uint8 | 告警类别 |
| AlarmObjType | Uint8 | 告警对象类型 |
| AlarmLevel | Uint8 | 告警级别 |
| ClearType | Uint8 | 告警清除类型：  0=自动清除，即必须由应用发起复位操作才能清除  1=手工清除，允许由客户端发起手工复位操作以清除告警 |

#### 其它告警配置

1. 故障管理许可配置：0=不许可平台管理故障，1=许可平台管理故障
   1. 当不许可平台管理故障时，所有满足AlarmFilter的告警均转发给另外的故障管理服务器【实际上作为一类客户端接入进来】。
2. 历史告警容量配置：历史告警，需要指定容量，当溢出时，将最先入历史告警的记录删除，在插入新恢复的告警信息。
   1. 当不许可平台管理故障时，也就没有历史告警了。
3. 故障服务器配置：
   1. 由于本平台让另外的故障管理服务器以客户端的形式接入进来，这里的配置将是客户端用户名列表。

### 事件数据

#### 事件数据模型

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 | 数据源 |
| EventNo | Uint32 | 事件编号 | 平台 |
| EventClass | Uint8 | 事件类别 | 应用（大类） |
| EventTime | Uint32 | 事件发生时间: | 应用 |
| EventObjType | Uint8 | 事件对象类型 | 应用（小类） |
| EventObjId | Uint8 | 事件对象标识 | 应用 |
| EventInfoLen | Uint16 | 告警信息长度，最长200字节 | 应用 |
| EventInfo | Uint8[] | 告警信息，长度为AlarmInfoLen | 应用 |

#### 事件配置

1. 故障管理许可配置：见4.1.2.5
   1. 当不许可平台管理故障时，所有事件均转发给另外的故障管理服务器【实际上作为一类客户端接入进来】。
2. 事件容量配置：需要指定容量，当溢出时，将最先入事件表的记录删除，在插入新的事件信息。
   1. 当不许可平台管理故障时，也就没有事件表信息了。
3. 故障服务器配置：见4.1.2.5

### 状态数据

#### 应用状态数据

不做数据模型的约定，在信令上定义一个通用的信令接口即可。

#### 平台状态数据

##### 状态数据

|  |  |  |
| --- | --- | --- |
| 数据项 | 数据类型 | 数据说明 |
| CPU | Uint32 | 单板CPU占用率，即CPU% |
| Memory | Uint32 | 单板内存占用率，即Memory% |

##### 配置数据

|  |  |  |
| --- | --- | --- |
| 数据项 | 数据类型 | 数据说明 |
| CPU\_Up | Uint32 | 超过该门限时，需要报告告警（一次） |
| CPU\_Down | Uint32 | 为了防抖，当超过上限门限后，再次跌下到下限门限时，才发清除告警。  当单板重启后，首先发送该清除告警。 |
| Memory\_Up | Uint32 | 超过该门限时，需要报告告警（一次） |
| Memory\_Down | Uint32 | 为了防抖，当超过上限门限后，再次跌下到下限门限时，才发清除告警。  当单板重启后，首先发送该清除告警。 |

### 业务管理数据

不做数据模型的约定，在信令上定义一个通用的信令接口即可。

### 系统目录的规划

该章节直接影响到应用管理、升级等业务流程和信令接口。

单板系统的大体启动流程：

1. 首先要连接到系统管理服务其中。
2. 下载一个需要下载的文件列表。
3. 本地进行MD5码校验。
4. 对MD5码校验不同的文件发起下载流程。
5. 启动应用。

这里要配置文件下载，但为了和升级管理等不冲突，最好和程序文件不要放在相同目录，以防软件升级时把配置文件覆盖掉。

应用系统在运行期间，可能还会动态生成并维护一些数据文件(非网管数据)，对于这类数据也最好不要和程序文件及网管数据文件冲突。

所以为了方便处理，本设计约定：单板系统分三大目录进行存放各类文件：

1. AppHome、用于存放应用软件(包括单板系统管理模块程序文件)
2. NmsHome、用于存放网管配置文件
3. RunHome、用于存放应用自身管理的文件（包括单板系统管理模块自身所管理的文件）,例如4.1.7.4定义的操作日志文件。

这几个顶级目录可通过环境变量指示。

### 统一编码

#### 管理范围编码

使用MDN(Manage Domain Number)管理范围编号为系统管理服务器进行逻辑编号。约定0xFF为无效值。

一个系统管理范围允许多个系统管理服务器，但是都拥有相同的系统管理编号，即从逻辑上只有一个系统管理服务器。

*在实际项目或产品中可用框号或框类型进行编码，因为某类单板可能只能插在某类机框上。*

#### 单板系统编码

使用MUN(Uni-board System Number) 单板系统编号为单板系统管理模块进行逻辑编号。一个系统管理范围可管理多个单板系统管理模块。约定0xFF为无效值。

单板系统管理模块自身也是一个应用，也需要按照应用系统编号法则进行进一步编号【参考4.1.5.3】。

*在实际项目或产品中可用槽位号或单板类型进行编码，为什么要涉及到单板类型呢？因为可能某类应用只能运行在特定硬件单板上。*

#### 应用系统编码

使用SSN(Sub System Number) 子系统编号为某类应用进行逻辑编号。同一单板可管理多类应用，但约定不能同时运行某类应用的多个实例。约定0xFF为无效值。

某类应用可以同时运行在不同的单板上【可属于相同或不同的管理范围】。

SSN是一种系统资源，需要集中统一管理，每开发一个新的应用，都需要申请一个SSN，**预定义SMS的SSN为0，SMU的SSN为1**，其它应用后续定义。

#### 管理对象编码

我们用OMI(Object Managed Identity) 管理对象标识为管理对象进行编码。根据数据模型的设计，我们可用表名来代表管理对象标识。并且根据表的树状特征，那么OMI也需要能表达层次关系。

在考虑到实际的网络部署，我们可以MDN-MUN-SSN-OMI来表示一个实际部署的管理对象，并用MMS(MDN-MUN-SSN)来表示管理对象的位置编码。

对于客户端而言MMS可以由系统管理服务器自行定义，对对于系统管理服务器和单板系统管理模块必须是明确的MDN-MUN-SSN语义。

## 通信模型

### 组网模型

#### 子网划分



这里以系统管理服务器为中心，把涉及到与之通信的网络实体的集合定义为子网。

根据管理范围SMD(System Manage Domain)的定义，我们划分两个子网，管理范围内的子网叫内子网，跨管理范围的子网叫外子网。外子网包含所有内子网的系统管理服务器节点。

单板系统(Uni-System)和客户端均仅具有一个子网，系统管理服务器具有两个子网。

子网划分的目的：

(A).客户端的安全访问和正确寻址（参考3.6章节）。

(B).减少数据同步量（参考3.3.1章节）。

(C).访问的简单性：

内子网的系统管理服务器对单板系统和客户端而言是一个逻辑实体，实际上会落在某一个系统管理服务器上，我们把这个服务器称为**活动系统管理服务器。**

由于有多个管理范围，上述机制仅仅降低了单板系统的接入难度，对于客户端而言并没有，所以我们再进一步在外子网上再竞选出一个**全局系统管理服务器，**作为客户端的统一接入实体。为了降低客户端对**全局系统管理服务器**的访问负荷，客户端仅从**全局系统管理服务器**获取当前的**活动系统管理服务器**列表**，**客户端随机选择一个进行访问（具体细节请参考3.6章节）。

外子网中，也不是所有系统管理服务器都能与客户端连接。换句话说外子网中还可以包含一个客户端的接入子网。这里不去划分单独的子网，我们将在虚拟IP设计章节屏蔽这个问题。

#### 管理IP划分

##### 客户端的管理IP

可用主机IP地址，即C-PIP。

##### 单板系统的管理IP

也可用主机IP地址，即B-PIP。

##### 系统管理服务器的管理IP

由于系统管理服务器包含两个子网，这两个子网可能在物理网络上是相通的，也可能是隔离的(如双网卡隔离)。这里进一步区分这两种情况。

###### 内外子网在同一网络平面



每个系统管理服务器有一个物理IP，即S-PIP。

为全局系统管理服务器提供一个VIP，即G-VIP，供客户端首次连接使用。请参考3.6 E-IF1接口。

为每个活动的系统管理服务器提供一个VIP，即S-VIP，用于系统管理服务器与单板系统及客户端的互联，请参考3.2 I-IF1接口和3.7 E-IF2接口。

所有的S-VIP、G-VIP、S-PIP均要求不同。

###### 内外子网在不同网络平面



为了安全，可用双网卡实现物理隔离，形成不同网络平面。那么系统管理服务器就会有两个物理IP，即S-I-PIP和S-E-PIP，其中S-I-PIP用于对内提供信令承载服务，S-E-PIP用于对外提供信令承载服务。

S-I-PIP用于管理范围内的系统管理服务器互联；S-E-PIP用于管理范围之间的系统管理服务器互联。

为全局系统管理服务器提供一个VIP，即G-VIP，供客户端首次连接使用。请参考3.6 E-IF1接口。G-VIP与S-E-PIP绑定在相同网卡设备上。

为活动的系统管理服务器提供一个内部VIP，即S-I-VIP，用于系统管理服务器与单板系统的互联，该S-I-VIP与S-I-PIP绑定在相同网卡设备上。

为活动的系统管理服务器提供一个外部VIP，即S-E-VIP，用于系统管理服务器与客户端的互联，该S-E-VIP与S-E-PIP绑定在相同网卡设备上。

如果内子网和外子网的网络是隔离的(双网卡可实现)，那么系统管理服务器就会有两个IP，一个IP用于系统管理服务器与单板系统之间的互联，另一个IP用于系统管理服务器之间互联以及系统管理服务器与客户端之间的互联。

所有的S-E-VIP、G-VIP、S-E-PIP不能相同，同一范围内的S-I-VIP、S-I-EIP不能相同。所有的S-I-PIP可以相同(建议)，所有的S-I-VIP可以相同(建议)。同一管理范围之内的单板物理IP不能相同。

###### 融合方案

（1）需要一个配置选项，指示网络组网方案，以区分是同一网络平面还是不同的网络平面。

（2）按内外子网在不同网络平面作为基础的设计方案，应对同一网络平面情况是，可按照下述方法解决：S-I-PIP取用S-E-PIP，S-I-VIP取用S-E-VIP相同。

（3）这些IP地址，不能通过网管配置，可通过其它途径(如DHCP)方式或按照某种规则计算获取。

##### IP绑定建议

针对VxWorks系统，一般系统启动时，是没有IP地址的，都是在启动过程中通过配置绑定的。

对于Linux系统，往往记住了应用绑定的IP地址，这对于IP地址的管理反而不方便。当在从网管上修改某个IP地址时，单板系统在修改前，发生异常而导致板卡重启，重新启动后，网管的配置已经修改，而单板应用不知道以前的IP地址，无法清除以前的IP地址。最小的问题，会导致单板IP地址垃圾，大问题是单板IP地址冲突。

可在Linux启动时做一个内存盘，把关于所有无需保存的文件放在该盘上【包括记录该IP地址的文件】，那么系统重启时所有IP地址都是空的，可用应用进行重新绑定。另外一种做法是：

1. 获取IP地址列表。
2. 清除系统现有的所有IP地址。
3. 重新绑定IP地址。

总之，单板出厂时需要能有一个默认IP，以便接入系统，获取真实的获取IP地址列表。默认IP可通过DHCP服务获取一个内网用的默认IP。

### 可靠通信

#### 网络承载协议选择

有很多协议可供选择，如TCP、UDP、SCTP、IPX、SPX等网络协议。为了降低系统要求，建议在最普通的TCP和UDP之间选择【其它协议需要特殊的软件包支持】。

选择TCP的好处在于自动实现传输的可靠性，不好的地方在于需要管理连接，针对每个连接需要独立的接收线程，比较浪费资源，当然也可以采用异步机制，但处理过程比较复杂。

数据同步这块，肯定需要选择多播UDP，TCP是不行的（因为需要两两建立连接，这个模型很丑陋），选择单播UDP的话，需要在不同UDP上发送数据，复杂度增大。既然选择了多播UDP，就得考虑消息的可靠性传输。

那么在一对一的通信中，如何选择呢？如果在多播的可靠性传输设计上兼容单播UDP的可靠性传输，那么单播UDP也是一个不错的选择。

**所以最终决策：全面采用UDP协议。**另外在一对一通信中，可以很容易设计出类似于SCTP的偶联特征，即：‘A连B’和‘B连A’完全等价。TCP是做不到这点的，TCP必须是两条双向链路，SCTP中是一条双向链路。

#### 消息序列化设计

##### 序列化模型

###### 组播序列化模型



###### 单播序列化模型



##### 序列化队列

对网络上接收到的消息按照接收的时序进行排队，是一个循环队列。上图其中0表示该序号对应的消息丢失，1表示该序号对应的消息未丢失，2表示再也取不回来消息了(将忽略处理)。

消息入队列机制，如果是新消息，放置队列尾端t，t向右移动。如果是旧消息，则填入相应位置（由滑窗控制识别消息的新旧）。

消息出队列机制，用上图说明，当头端h空消息处填充好后，立即启动消息出队列流程，弹出一个处理一个，h向右移动，知道h到达t或空消息位置。

消息的入队列和出队列由同一线程完成。

序列化消息队列节点记录消息的内容和发端节点编号。

##### 接收滑窗

对某个节点的接收消息进行按消息序号排队，滑窗节点项记录消息属性：(1)是否为空，(2)对应在序列化队列的位置。

接收端滑窗记录头/尾端的消息序号。接收到某条消息，需要判断该消息是否过期【丢包时间太长，启动过异常处理机制】，还需要判断某条消息是否为新消息，若是新消息，与尾端消息相差多少个丢包消息。

当序列化队列头端右移时，根据该消息发端节点编号，对相应接收端滑窗进行头端右移。

###### 滑窗重要参数

nSize，表示滑窗大小。

nHead，表示滑窗的头部消息序号。取值范围：[ 0，nSize-1 ]。

nTail，表示滑窗的尾部消息序号(下一个应收的消息)。取值范围：[ 0，nSize-1 ]。

bFirst，表示系统没有放入过消息。

滑窗存储的消息范围为：[ nHead，nTail )。

初始：nHead = 0；nTail = 0；bFirst=true；

收到第一个序列化消息时，会执行下述逻辑。

if(bFirst)

{

nHead = nSeq;

nTail = nSeq;

bFirst = false;

}

###### 消息存放位置算法

nDistance = nSeq % nSize

其中nSeq表示当前消息的序号。

nDistance表示应该存储在滑窗中的位置，如果用数组实现滑窗oWindow，那么oWindow [nDistance]就是代表消息存放所在的内存空间。

###### 消息过期算法

nDistance1 = nSeq - nHead;

if(nDistance1 >= nSize)

{

if(nDistance1 <=( 0xFFFFFFFF - nSize + 1))

{

//可能是系统重启或长时间断链导致。

//滑窗异常处理，继续后续流程。

}

else

{

//消息过期，可丢弃当前消息，中断当前流程。

}

}

###### 新消息与旧消息的判断

nDistance2 = nTail - nHead;

if(nDistance1 >= nDistance2)

{

//表示nSeq对应的消息是新消息

//nSeq - nHead表示应该插入的空消息数。

//如果空消息插入序列化队列失败(满了)，需要启动滑窗异常处理。

}

else

{

//表示nSeq对应的消息是旧消息

//nSeq% nSize表示消息应该存于何处。

}

###### 滑窗异常处理

while(nHead!=nTail)

{

nDistance = nHead % nSize

if(oWindow[nDistance]是空消息)

{

//那么在序列化队列的相应位置也应该是空消息

//应该认为该空消息再也取不回来了，设置为2，让系统继续能处理。

}

++ nHead;

}

bFirst = true;//很重要，表示下次收到消息，将重新开始。

除了消息序号在可承受范围内过期需要进行滑窗异常处理外，收到注销消息，也需要进行滑窗异常处理。

##### 保持队列

当开启序列化服务后，允许接收序列化消息，但是由于某种原因，致使上层应用模块还不能真正成为服务状态时，序列化消息出队列后，需要放入保持队列中，当上层应用模块真正成为服务状态时，才开始序列化消息的处理。

直到保持队列为空时，序列化消息即转入即时处理。当然该队列是可选的。

##### 发送队列

###### 组播发送队列

发送端队列是一个循环队列，队列满时循环覆盖。

###### 单播发送队列

发送队列中的消息分两类：（1）群发消息，（2）非群发消息。使用目标节点号来分别，-1表示群发消息，否则为具体目标的非群发消息。

由于有非群发消息，所以为每个目标节点单独维护消息序号，用发送序号队列表示（类似于接收窗口）。

实际设计时，需要支持两种模式，一种仅支持群发的模式，一种支持非群发模式。

##### 消息接收流程

收到的消息，从消息中提取发送方身份NodeId，并将序号信息放入相应的接收滑窗。

(a)过期包(序号超出队列范畴)，则丢弃。

(b)重包(序号已在队列中，且有非空标记)，则丢弃。

(c)丢包(序号已在队列中，且有空标记)，则根据位置信息，直接把消息数据存入序列化队列。

(d)正常包(序号不在队列中，且未超出队列范畴)：

(i)该序号与节点队列的最后一个消息的序号有空隙，表示有丢包，则插入空消息到序列化队列(返回位置信息)，并将位置信息存入节点队列，做空标记。

(ii)若无丢包，将消息存入序列化队列(返回位置信息)，并将位置信息存入节点队列，做非空标记。

##### 消息重传流程

定时扫描接收端滑窗，提取空消息标记，用单播模式向该节点发送端发送消息重传请求。发送端收到消息重传请求时，从发送队列提取相应消息，并用单播模式重发该消息。

##### 异常流程

异常发生时，序列化队列将从头到尾的处理消息【忽略空消息】。有以下几种异常情况：

(A)，接收端滑窗或序列化队列满。

(B)，发现某个节点重新注册，而该节点的接收滑窗还有消息。

(C)，某节点长时间断连。

再好的序列化方案都只能保证一定时间网络断链情况，网络恢复正常后，系统的数据能一致起来。

长时间断链情况，可作为特殊情况对待，演进版本可考虑一致性检查和恢复机制。为预留这个机制，我们将在数据模型的设计上增加时间信息，一致性恢复时将以最新时间为依据。一致性检查出差异报告，可支持自动恢复和人工恢复两种模式，自动恢复的原则是：即便数据错误，也要让整个系统的数据错成一样的。

##### 定时流程

有三个定时流程。

(A)定时注册，建议1秒钟一次注册。

(B)定时检查丢包情况，发起消息重传流程。

(C)定时断线检测，发起异常处理流程。

##### 序列化能力

支持单播序列化和多播序列化两种能力，这两种能力不能在一个服务对象中同时支持。

在系统中要同时支持两种能力，就需要创建两个序列化对象。

###### 多播序列化能力

多播序列化能力分为：序列化接收和发送两种能力，这两种能力可以同时并存。

如果支持序列化接收，如果链路的对端有发送能力，那么就含有接收滑窗。如果不需要支持序列化接收能力，那么收到序列化数据消息时，直接丢弃。

如果支持序列化发送，就含有发送队列。

如果两个能力都不提供，是不能建立多播序列化服务的

###### 单播序列化能力

单播序列化能力：

序列化接收能力

序列化发送能力：

序列化可靠发送能力

序列化非可靠发送能力

序列化接收和发送两种能力是可以同时并存的，但‘序列化可靠发送能力’和‘序列化非可靠发送能力’是二选一的。

如果支持序列化接收，如果链路的对端有发送能力，那么就含有接收滑窗。如果不需要支持序列化接收能力，那么收到序列化数据消息时，直接丢弃。

如果支持序列化发送，就含有发送队列。如果支持序列化可靠发送能力，会为每个具备接收能力的链路创建发送窗口。否则仅创建一个统一的窗口。

###### 链路状态

|  |  |
| --- | --- |
| 状态 | 状态描述 |
| OnLine | 在线状态 |
| Break | 断线状态 |
| OffLine | 离线状态，离线状态下，我们会将链路删除。 |

#### 令牌模型设计

通信模块子网中有时为了某种需要，需要独占时间，进行某种事务的处理。本令牌设计模型实现1+m的能力，为了支持n+m的能力，我们设计多令牌支持方案，即n个令牌就实现了n+m的能力，这里n+m总数就是子网节点数。

本节所描述的设计方案依据RFC5059标准。

##### 令牌结构设计

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| TokenId | Uint32 | 令牌标识。 |
| DefaultNode | Uint32 | 缺省由那个节点占用。 |
| Nodes | strings | 用逗号分隔的参与竞选的节点列表。 |
| SDN | uint32 | 令牌竞选的同步范围。 |

令牌是一种资源，就像网络端口号一样，其编号必须统一定义和管理，以防不同模块之间的冲突。

##### 令牌对应用的接口。

令牌是一个自动化服务，根据配置自动启动。

采用回调接口告知应用某个令牌被抢占或放弃。

##### 令牌竞选流程



其中四个节点CPEA表示四个竞选状态（P为初始状态），有向连线表示状态迁移的驱动消息。

###### 竞选状态

|  |  |
| --- | --- |
| 状态 | 状态描述 |
| C | Candidate 候选状态。本节点是候选者。此时另外一个节点是占有令牌。 |
| P | Pending 等待状态。等待选举结果，此时本节点认为无其他节点占有令牌，但本机还未当选。 这个临时状态是为了防止竞选时出现振荡。 |
| E | Elected挂职状态。节点已当选，但未正式就职的准备状态，这个状态是为了原令牌节点做清理。 |
| A | Acknowledged就职状态。表示本节点正式占有令牌。 |

###### 驱动消息

p=Preferred BSM; 更优BSM消息

n = Non-Preferred BSM非最优BSM消息

n\*= Non-preferred BSM from Elected Owner当选Owner的非最优BSM消息

T= Bootstrap Timer timeout超时消息，不同状态下timer值可不同。

BSM(Bootstrap Message)消息定义：其实只有1条消息，但是根据消息发送者的weight和接收者的状态、接收者自己的weight、以及当前Owner的weight的对比关系，可以分为（每个节点都要记录当前Owner的地址信息和weight）：

P状态： 若‘发送者权重’> ‘自身权重’，则‘P消息’，否则‘N消息’

A、E状态：若‘发送者权重’> =‘Owner权重’，则‘P消息’，否则‘N消息’

C状态：

（1）若‘发送者权重’> =‘Owner权重’，则‘P消息’，否则：

（2）若‘发送者权重’> =‘自身权重’，则‘P消息’，否则：

（3）若‘发送者 == Owner’，则‘N\*消息’，否则‘N消息’

###### 状态签转

C候选状态 (Candidate)

1. 收到Preferred BSM 消息：

* 状态不变。
* 存储Owner信息；
* 设置BST定时器为 BS\_timeout

1. 收到BST超时消息：

* 转P状态，老Owner没动静了，我要参选！
* 设置BST定时器值为 BS\_rand\_override。

1. 收到n\*消息

* 转P状态 ，老Owner体力不支了（weight变小了，这种情况下其他成员也都知道了），或者本机自身的weight提高了（这种情况只有本机自己知道），我要参选！
* 设置BST定时器值为 （BS\_timeout + BS\_rand\_override）。

1. 收到n：（RFC中没有定义处理行为）

* 状态不变

P等选状态（Pending）

1. 收到Preferred BSM 消息：

* 转C状态
* 存储Owner信息；
* 设置BST定时器为 BS\_timeout

1. 收到BST超时消息（BS\_timeout）：

* 转E状态，当选，但不正式‘就职’
* 发送 BSM消息。
* 设置BST定时器值为 BS\_prepare。

1. 收到Non-preferred BSM：（RFC中没有定义处理行为）

* 状态不变。

E挂职状态 (Elected)

1. 收到Preferred BSM 消息：

* 转C状态 – 比自己更优的组长来了，立刻让位…
* 存储Owner信息；
* 设置BST定时器为 BS\_timeout

1. 收到BST超时消息：

* 转 A状态，Acknowledged，正式就职
* 发送 BSM消息
* 设置BST定时器值为 BS\_period。
* 将自身信息保存为Owner信息，并对应用产生竞选成功的回调。

1. 收到Non-preferred BSM：

* 状态不变
* 发送 BSM消息，有挑战者立刻镇压
* 原BST定时器不重置，继续计时

A就职状态 (Acknowledged)

1. 收到Preferred BSM 消息：

* 转C状态 – 比自己更优的组长来了，立刻让位…
* 存储Owner信息；
* 设置BST定时器为 BS\_timeout
* 对应用产生放弃令牌的回调。

1. 收到BST超时消息：

* 状态不变
* 发送BSM消息--该提醒一下各位了，我是组长
* 设置BST定时器值为 BS\_period

1. 收到Non-preferred BSM：

* 状态不变
* 发送BSM消息--不知轻重的小卒来挑战，立刻镇压

###### 竞选权重

每个节点对某个令牌设计的权重计算如下：

1. 基础权重b-weight：当前节点是该令牌的缺省节点，那么b-weight=192，否则b-weight=64
2. 参考权重r-weight：用当前节点的节点号作为参考权重。
3. 节点权重n-weight(uint64) =( ((uint64)b-weight)<<32) | r-weight。

###### 竞选定时

|  |  |  |
| --- | --- | --- |
| 编号 | 名称 | 说明 |
| T0 | bs\_refer | 参考时长（以秒为单位） |
| T1 | bs\_period | 当选的Owner周期性宣告。1.5\*T0 >= T1 >= T0 |
| T2 | bs\_timeout | 成员探测Owner不可达时长。建议T2=2\*T1+ T0 = 4\*T0 |
| T3 | bs\_prepare | 挂职等待时长，该时间用于原Owner做清理工作。建议为1.5\*T0 |
| T4 | bs\_rand\_override | 从C到P后设置的定时时长。weightDelay+ addrDelay+ T0。  weightDelay = (MaxWeight – b\_weight) 0.5\*T0/10  addrDelay = myIPAddress的第三字节 / 128\*T0/10 |

换句话说竞选定时仅需要一个参考时长的配置，该时长在测试阶段可作为临时配置，系统稳定后，根据测试情况固定一个值即可。

###### 默认调整

当网管发起命令进行默认抢占令牌的节点标识变化时，

1. 当自己之前是默认的，本调整为非默认的，需要更新。
2. 当自己之前不是默认的，调整为默认的，也需要更新。
3. 否则不做任何更新。

当自己的默认角色发生改变时，需要重新计算权重，和对该令牌的T4定时时长。

##### 异常处理

当网络出现异常时，可能会引发多个子网的出现，从而导致一个令牌同时被多个节点占用。这个问题并不会引起太大的影响，只要当网络恢复正常时，会有一个收敛时间（>=T2），该收敛时间一过，是可以重新竞选出新占用者的。

#### 虚拟IP设计

这里的虚拟IP，实际上也是真实的IP地址。基于令牌技术，获得令牌的节点可以绑定相应IP地址，失去令牌的节点需要解除相应IP地址的绑定。

4.1.2章节中所涉及到的VIP，可以采用该方案进行。

特别需要提示的是G-VIP，并不是所有系统管理服务器都能与客户端连接，前面讲到我们没有划分单独的子群，我们可以在令牌的参与竞选者列表中做约束，把能与客户端连接的系统管理服务器放置到G-VIP对应的令牌竞选者列表中即可。

#### 内部信令定义

信令采用分层设计方式，这是为了：

1. 做成一个较为通用的信令模块，也便于以后的扩展。
2. 屏蔽应用不关心的内部信令。
3. 对命令字进行分段，那么各个段的命令字互不冲突。

目前分三层：

|---顶层，定义基本的UDP信令。

|---序列化信令---实现可靠传输

|---大包信令。

|---基本文件传输信令。

|---其它上层扩展的信令。

|---令牌信令（不能可靠传输，所以不能和‘序列化信令’混为一谈，故需要分层）。

##### 最大消息长度

由于采用UDP协议，需要定义最大消息长度，默认为4096(MAX\_UDP)。

##### 基本信令定义【12字节】

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| NodeNo | Uint32 | 发送节点编号： |
| PidNo | Uint32 | 发送进程编号，为了踢出重复消息。 |
| Module | uint16 | 信令服务模块号。目前预留定义【可扩展】：  0=序列化服务模块号。  1=令牌服务模块号。 |
| Cmd | uint16 | 描述消息命令字。由服务模块自行定义。 |
| Body | uint8[] | 由于是UDP消息，不做消息体长度参数。 |

PidNo的作用：

1. 收到节点号重复的消息，有两种情况：
   1. 由于组播回环【特定情况下许可】，收到自己发出的消息，是合理的，直接丢弃即可。
   2. 收到其它其它错误配置节点发出的消息，是不合理的，需要分下述两种情况分别处理：
      1. 如果系统在正常运行中，收到后来者的不合理消息，直接丢弃即可【需要保护当前正常运行的系统】
      2. 如果系统在启动过程中，收到已有节点的不合理消息，应该认为当前节点配置不正确，不允许系统启动。
2. 那如何判断是不合理消息呢？【节点号相同情况下】
   1. 如果是单播模型，由于没有回环概念，可直接认为消息不合理。
   2. 如果是组播模型，就有回环的情况，可按下述两种情况依次判断：
      1. 如果发送方的IP地址和当前节点的IP地址不同，可判断消息不合理。
      2. 如果发送方的进程号和当前节点的IP进程号不同，可判断消息不合理。

**注意本协议中使用的string类型的字段，若无特殊说明，均以0为结尾进行传输。**

##### 序列化信令定义

在基本信令定义的基础上定义四个序列化信令命令字的定义【序列化服务模块】：

|  |  |  |
| --- | --- | --- |
| 命令字 | 命令号 | 命令说明 |
| SeqRegister | 0 | 序列化模块注册信令 |
| SeqDeRegister | 1 | 序列化模块注销信令 |
| SeqData | 2 | 序列化模块数据信令 |
| SeqResendReq | 3 | 序列化模块重传信令 |
| SeqUcAck | 4 | 序列化模块单播确认信令 |
| SeqResendData | 5 | 序列化模块重传数据信令，结构与SeqData一致，当数据段为空时表示空消息，即指示SeqResendReq的失败响应 |

(1) SeqRegister消息体定义(定时发起)

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| nTime | uint32 | 程序启动时间，以秒为单位，以此判断是首次注册还是后续握手。 |
| nHost | uint8[4] | 描述消息发送方主机地址，重传单播地址。 |
| nUniCast | Uint8 | 0=多播；1=单播；  注意单播节点不能向多播节点注册，反之亦然。 |
| nStatus | Uint8 | 描述当前的服务状态。0=初始状态；1=服务状态； |

(2) SeqDeRegister消息体定义 (退出发起)

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |

注销时(如系统退出)，发送该消息。如果是单播模型，需要扫描接收端节点列表，逐一发送该消息。

(3) SeqData消息体定义【12+4字节】

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| Sequence | uint32 | 消息序号，发送方维护该序号的唯一性，循环递增即可。 |
| Data | uint8[] | 序列化消息包数据。Data前四个字节为序列化操作SeqOp，其它数据为序列化数据。  **模块提供序列化消息处理函数的注册，同时预留1000以内的命令字给系统自身使用。** |

序列化数据SeqOp结构定义：

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| PlugIn | uint16 | 信令服务插件号。目前预留定义【可扩展】：  0=分包插件 |
| Cmd | uint16 | 描述消息命令字。由服务模块自行定义。  分包插件的命令字定义：  0=大的信令包  1=大的数据流，如文件传输可用这个。 |

超长包的拆包和组包机制：

1. 分包Data结构【12+4+6】

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| SeqOp | Uint32 | 0，分包SeqOp |
| PackNo | uint16 | 分包的包号。 |
| Data | uint8[] | 原包数据，PackNo为0的头4字节是SeqOp |

1. 拆包原理：
   1. 如果原包数据长度大于MAX\_UDP -16字节，需要拆包。
   2. 每个分包的数据最长MAX\_UDP -22字节。
2. 自动组包原理：
   1. 设置一个组包是否失败的标记，初始为失败。
   2. 收到包号为0，如果之前的组包没有失败，则处理整个消息。清除组包错误标志，并重新开始组包。
   3. 如果出现包号不连续，标记组包失败，直到包号为0。
3. 手工组包原理【模仿流的概念】：
   1. 设置一个组包是否失败的标记，初始为失败。
   2. 收到包号为0时，回调应用读包函数，应用自行收包。
   3. 当包断序时，会返回一个接收结束的状态，直到包号为0。

(4) SeqResendReq消息体定义

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| Package | uint32 | 需要重传的包数量，一次性可以发起多个丢包重传请求。 |
| Sequence | uint32[] | 重传包的消息序号。 |

##### 令牌信令定义【12+8字节】

在基本信令定义的基础上定义一个令牌信令命令字的定义【令牌服务模块】：

|  |  |  |
| --- | --- | --- |
| 命令字 | 命令号 | 命令说明 |
| TokenBootStrap | 0 | 令牌宣告信令 |

TokenBootStrap消息体定义：

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 参数说明 |
| TokenId | uint32 | 竞选令牌标识。 |
| Weight | uint32 | 发送者权重。 |

## 架构建议

### 系统管理服务器架构设计建议



单播序列化模块同时管理客户端和单板系统，数据传输和前转信令都走该通道。S-I-VIP和S-E-VIP两个虚拟IP对应的令牌【只有一个令牌】在内子网多播上竞选。G-VIP对应的令牌在外子网多播上竞选。系统同时仅支持一条数据传输通道。

多播序列化模块分管理范围内的【内子网】和管理范围间【外子网】。

数据管理部件，除了要考虑本地内存、文件读写外，还要考虑数据库的存储。以及进行分布式的数据管理。如果配置部分第一阶段想做的很简单，可以仅考虑故障事件的入库即通知处理。

大体启动流程：

(1)判断数据库系统的表格是否建立好，没有需要建立，SQL脚本需要考虑安全性。当没有表格时创建表格，当没有索引时建立索引等，还要考虑系统升级的字段扩展。

(2)加载本地Xml数据

先加载Xml文件，再加载操作日志。

(3)从网络子网中选择一个节点进行加载数据。【参考4.3.2.2】

(4)刷新Xml文件。

(5)设置处于服务状态。

启动一个独立的任务，处理XmlDbSync队列中的消息，当该队列为空时，转入XmlDbSync的即时处理。XmlDbSync需要做操作日志。

同时启动令牌竞选流程。

### 单板系统管理模块架构设计建议



单板系统要负责对业务系统的管理，由业务系统连接到单板系统。业务系统需要自行判断是否处于僵死状态，如果处于僵死状态，可以通过单播序列化模块主动发起注销消息。以便让单板系统管理模块感知业务系统的运行状态。

数据管理模块需要支持本地存储的数据组织[配置数据]。

大体启动流程：

1. 连接系统管理服务器(S-I-VIP)
2. 文件下载流程
3. 启动各应用[整个系统可能需要重启]

### 业务系统数据管理部件架构设计建议



数据管理模块需要考虑本地数据存储的解析，还需要考虑单板系统管理模块对业务系统的动态修改通知以及业务系统对数据的动态上报封装。

业务系统需要自行判断是否处于僵死状态，如果处于僵死状态，可以通过单播序列化模块主动发起注销消息。以便让单板系统管理模块感知业务系统的运行状态。

### 客户端数据管理部件架构设计建议



数据管理模块并不实现本地存储，仅需要实现消息及XML数据的解析和组包即可，通过单播序列化模块和系统管理服务器通信，完成各种交互(参考第3章相关章节)。

客户端向G-VIP发出GetServersRequest后，即进行串行等待。

客户端基于时间随机选择一个S-E-VIP，发送ClientLoginRequest给相应的活动系统管理服务器，即进行串行等待。

登陆成功后，启动可靠的序列化服务。

# 需求的可追踪性

## 部署需求

| **需求 ID** | **需求标题** | **对应章节** |
| --- | --- | --- |
| REQ-ARRAY-1 | 支持单域单网元部署 | 4.2.2.1 |
| REQ-ARRAY-2 | 支持单域多网元部署 | 4.2.2.1 |
| REQ-ARRAY-3 | 系统管理信令的路由 | 4.1.5，4.1.6，3.3 |
| REQ-ARRAY-4 | 系统管理平台保证系统管理信息在平台内传输的可靠性 | 4.2.2 |
| REQ-ARRAY-5 | 系统管理平台内的系统信息同步 | 3.3 |
| REQ-ARRAY-6 | 系统管理平台支持多客户端登录 | 3.6 |
| REQ-ARRAY-7 | 系统管理平台支持客户端同时多点连接 | 3.6 |
| REQ-ARRAY-8 | 系统管理平台内安全信息同步 | 3.3 |

## 数据配置功能需求

| **需求 ID** | **需求标题** | **对应章节** |
| --- | --- | --- |
| REQ-CONFIG-1 | 系统管理平台对外提供统一的配置数据访问与操作功能 | 4.1.6  3.6 |
| REQ-CONFIG-2 | 统一的系统管理配置处理逻辑，不受新增配置业务影响 | 4.1 |
| REQ-CONFIG-3 | 系统管理平台支持不同数据格式配置需求 | 4.1 |
| REQ-CONFIG-4 | 系统管理平台与客户端之间配置数据交互 | 4.1.6  3.6 |
| REQ-CONFIG-5 | 系统管理平台与业务应用之间配置数据交互 | 4.1.6  3.7 |
| REQ-CONFIG-6 | 系统管理平台支持XML数据解析 | 4.1 |
| REQ-CONFIG-7 | 系统管理平台具备配置数据存储功能 | 3.5 |
| REQ-CONFIG-8 | 系统管理平台支持业务配置数据同步 | 4.2.2  3.3.1 |
| REQ-CONFIG-9 | 系统管理平台支持配置数据校验 | 4.1.3 |
| REQ-CONFIG-10 | 系统管理平台支持配置数据回滚功能 | 由上层应用支持 |
| REQ-CONFIG-11 | 配置数据备份与恢复 | 由上层应用支持 |

## 故障管理功能需求

| **需求 ID** | **需求标题** | **对应章节** |
| --- | --- | --- |
| REQ-ALARM-1 | 故障的上报通道 | 由上层应用进一步支持，可以定义特定的故障表格，对于该协议接口，全部统一为表格操作。 |
| REQ-ALARM-2 | 故障信息定位功能 |
| REQ-ALARM-3 | 故障信息识别功能 |
| REQ-ALARM-4 | 故障信息保存及维护 |
| REQ-ALARM-5 | 故障信息检索 |
| REQ-ALARM-6 | 故障信息备份功能 |

## 业务管理功能需求

| **需求 ID** | **需求标题** | **对应章节** |
| --- | --- | --- |
| REQ-SERVICE-1 | 系统管理平台提供业务管理通道 | 4.1.3.2，采用‘控制类型’字段用于实现业务控制功能。 |

## 系统监控功能需求

| **需求 ID** | **需求标题** | **对应章节** |
| --- | --- | --- |
| REQ-MONITOR-1 | 系统管理平台支持对单板与进程进行监控 | 3.4.2定义“16=监控”，实现自动监控功能。 |
| REQ-MONITOR-2 | 系统管理平台监控功能上报故障信息 |
| REQ-MONITOR-3 | 系统管理平台支持对监控门限值进行配置 |
| REQ-MONITOR-4 | 系统管理平台支持对监控结果进行查询 |

## 系统信息上报功能需求

| **需求 ID** | **需求标题** | **对应章节** |
| --- | --- | --- |
| REQ-EVENT-1 | 事件信息上报 | 由上层应用进一步支持，可以定义特定的故障表格，对于该协议接口，全部统一为表格操作。 |
| REQ-EVENT-2 | 事件信息的保存与维护 |
| REQ-EVENT-3 | 事件信息检索 |
| REQ-EVENT-4 | 事件信息备份功能 |

## 软件管理功能需求

| **需求 ID** | **需求标题** | **对应章节** |
| --- | --- | --- |
| REQ-SOFT-1 | 系统管理平台软件管理功能 | 4.2.2.5.5提供FTP信令支持，  3.2.1提供应用升级、启动的基本信令支持。 |
| REQ-SOFT-2 | 系统管理平台自身软件故障处理 |
| REQ-SOFT-3 | 系统管理平台业务软件的启动管理 |
| REQ-SOFT-4 | 系统管理平台业务软件启动过程管理 |
| REQ-SOFT-5 | 系统管理平台监控业务软件运行 |
| REQ-SOFT-6 | 系统管理平台处理业务软件运行异常 |
| REQ-SOFT-7 | 系统管理平台支持软件管理策略配置 |
| REQ-SOFT-8 | 系统管理平台支持软件结束通知 |
| REQ-SOFT-9 | 系统管理平台支持文件下载机制 |
| REQ-SOFT-10 | 系统管理平台支持业务软件的更新 |
| REQ-SOFT-11 | 系统管理平台支持客户端对业务软件运行控制功能 |

## 系统安全功能需求

| **需求 ID** | **需求标题** | **对应章节** |
| --- | --- | --- |
| REQ-SAFE-1 | 系统管理平台提供对用户权鉴机制 | 3.4.1 |
| REQ-SAFE-2 | 系统管理平台支持灵活的用户权鉴方式 | 3.4.2 |
| REQ-SAFE-3 | 系统管理平台灵活的用户权限设定 | 3.4.2 |
| REQ-SAFE-4 | 系统管理平台的用户权限告知 | 3.4.1 |
| REQ-SAFE-5 | 客户端登录后的信令安全保证 | 3.4.3 |
| REQ-SAFE-6 | 用户操作日志自动记录 | 由上层应用支持 |
| REQ-SAFE-7 | 系统管理平台支持安全配置 | 由上层应用支持 |

# 注释

|  |  |
| --- | --- |
| 缩略语 | 原语及说明 |
| NMS | Network Manage System，网络管理系统 |
| SNMP | Simple Network Manage Protocol，简单网络管理协议 |
| TR069 | 基于SOAP的远端设备管理协议，不仅仅是通信协议，同时也定义了管理模型。 |
| SMP | System Manage Platform，系统管理平台 |
| SMS | System Manage Server，系统管理服务器 |
| SMC | System Manage Client |
| SMU | uni-board System Manage Unit，单板系统管理模块 |
| SMD | System Manage Domain，系统管理范围 |
| MDN | Manage Domain Number，管理范围编号 |
| MUN | Manage Unit Number，管理模块编号 |
| SDN | Synchronization Domain Number，同步范围编号 |
| SSN | Sub System Number，子系统编号 |
| OMI | Object Managed Identity管理对象标识 |
| MMS | MDN-MUN-SSN，管理对象位置标识。 |

阴影部分是标准命名，其它是本协议接口定义的私有命名。