

ICS 33.040

M 15

YD

中华人民共和国通信行业标准

YD/T 1878.3-2009

800MHz/2GHz cdma2000 数字蜂窝移动通信网

高速分组数据 (HRPD) 网络管理技术要求

第3部分：基于 CORBA 技术的网络资源模型设计

800MHz/2GHz cdma2000 Digital Cell Mobile Communications Network
(HRPD) Network Management Technical Specification

Part 3: Network Resource Model Design based CORBA

2009-06-15 发布

2009-09-01 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 缩略语	1
4 配置网络资源模型设计	1
4.1 通用配置资源模型的 IDL 定义	1
4.2 无线接入网网络资源模型的 IDL 定义	16
4.3 核心网网络资源模型的 IDL 定义	21
5 性能网络资源模型设计	26
5.1 性能参数的 IDL 定义	26
5.2 数据类型的 IDL 定义	31
6 性能管理接口功能相关的文件	34
6.1 性能测量数据文件的 Schema 定义<HrpdMeasCollec.xsd>	34
6.2 性能测量数据文件的 XML header 定义	41
附录 A (规范性附录) XML Schema 文档补充说明	42
附录 B (资料性附录) 性能管理功能相关 XML 文件示例	44
参考文献	48

前 言

《800MHz/2GHz cdma2000 数字蜂窝移动通信网高速分组数据（HRPD）网络管理技术要求》分为 3 个部分：

- 第 1 部分：配置网络资源模型
- 第 2 部分：性能网络资源模型
- 第 3 部分：基于 CORBA 技术的网络资源模型设计

本部分为第 3 部分。

附录 A 为规范性附录，附录 B 为资料性附录。

本部分由中国通信标准化协会提出并归口。

本部分起草单位：北京邮电大学、中兴通讯股份有限公司、北京市天元网络技术股份有限公司、中国联合网络通信有限公司

本部分主要起草人：芮兰兰、李文璟、王智立、高志鹏、朱 凯、黄 波、苏晨阳、郑兴明、梁 亮、王 勇

广东省网络空间安全协会受控资料

800MHz/2GHz cdma2000 数字蜂窝移动通信网

高速分组数据（HRPD）网络管理技术要求

第3部分：基于 CORBA 技术的网络资源模型设计

1 范围

本部分规定了800MHz/2GHz cdma2000数字蜂窝移动通信网高速分组数据（HRPD）的网络管理接口的CORBA/IDL定义。

本部分适用于800MHz/2GHz cdma2000数字蜂窝移动通信网高速分组数据（HRPD）的网络管理。

2 规范性引用文件

下列文件中的条款通过本部分的引用而成为本部分的条款。凡是注日期的引用文件，其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分。然而，鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本部分。

YD/T 1878.1-2009 800MHz/2GHz cdma2000数字蜂窝移动通信网高速分组数据（HRPD）网络管理技术要求 第1部分：配置网络资源模型

YD/T 1878.2-2009 800MHz/2GHz cdma2000数字蜂窝移动通信网高速分组数据（HRPD）网络管理技术要求 第2部分：性能网络资源模型

3 缩略语

下列缩略语适用于本部分。

IDL	Interface Definition Language	接口定义语言
CORBA	Common Object Request Broker Architecture	公共对象请求代理体系

4 配置网络资源模型设计

注：配置网络资源模型设计中有3类idl文件,这3类文档及其用途如下：

1) xxxNRMDefs.idl包括GenericNRMDefs.idl、IMDataDefs.idl、RanNRMDefs.idl和CoreNRMDefs.idl，用来定义配置网络对象及其属性名称；

2) xxxNRMSystem.idl包括GenericNRMSystem.idl、RanNRMSystem.idl和CoreNRMSystem.idl，用来定义配置网络资源对象的属性使用的数据类型；

3) xxxNRMPProfile.idl包括GenericNRMPProfile.idl、IMDataProfile.idl、RanNRMPProfile.idl和CoreNRMPProfile.idl，只是用来描述配置网络资源对象的属性名称及其数据类型的对应关系，实现时并不使用此类IDL文件。

4.1 通用配置资源模型的 IDL 定义

4.1.1 GenericNRMDefs

```
//File "GenericNRMDefs.idl"
//The IRP document version number is "Generic NRM V1.1"
#ifndef _GENERIC_NRM_DEFS_IDL_
```

```
#define _GENERIC_NRM_DEFS_IDL_

//This module defines constants for each MO class name and
//the attribute names for each Generic MO class.

module GenericNRMDefs
{
    //Definitions for abstract MO class Top

    interface Top
    {
        const string ObjectClass = "ObjectClass";
        const string ObjectInstance = "ObjectInstance";
    };

    //Definitions for MO class IRPAgent

    interface IRPAgent: Top
    {
        const string CLASS = "IRPAgent";

        // Attribute Names
        //
        const string iRPAgentId = "iRPAgentId";
        const string systemDN = "systemDN";
    };

    //Definitions for abstract MO class GenericIRP

    interface GenericIRP: Top
    {
        const string CLASS = "GenericIRP";

        // Attribute Names
        //
        const string iRPId = "iRPId";
    };

    //Definitions for MO class SubNetwork

    interface SubNetwork: Top
    {
        const string CLASS = "SubNetwork";

        // Attribute Names
```

```
//
const string subNetworkId = "subNetworkId";
const string dnPrefix = "dnPrefix";
const string userLabel = "userLabel";
const string setOfMcc = "setOfMcc";
const string userDefinedNetworkType = "userDefinedNetworkType ";
};

//Definitions for MO class MeContext

interface MeContext: Top
{
    const string CLASS = "MeContext";

    // Attribute Names
    //
    const string meContextId = "meContextId";
    const string dnPrefix = "dnPrefix";
};

//Definitions for MO class ManagementNode

interface ManagementNode: Top
{
    const string CLASS = "ManagementNode";

    // Attribute Names
    //
    const string managementNodeId = "managementNodeId";
    const string managedElements = "managedElements";
    const string userLabel = "userLabel";
    const string userDefinedState = "userDefinedState";
    const string swVersion = "swVersion";
    const string locationName = "locationName";
    const string vendorName = "vendorName";
};

//Definitions for MO class ManagedElement

interface ManagedElement: Top
{
    const string CLASS = "ManagedElement";

    // Attribute Names
    //
```

```
const string managedElementId = "managedElementId";
const string dnPrefix = "dnPrefix";
const string userLabel = "userLabel";
const string vendorName = "vendorName";
const string locationName = "locationName";
const string managedElementType = "managedElementType";
const string managedBy = "managedBy";
const string userDefinedState = "userDefinedState";
const string swVersion = "swVersion";
};

//Definitions for abstract MO class ManagedFunction

interface ManagedFunction : Top
{
    const string CLASS = "ManagedFunction";

    // Attribute Names
    //
    const string userLabel = "userLabel";
};

//Definitions for MO class VsDataContainer

interface VsDataContainer: Top
{
    const string CLASS = "VsDataContainer";

    //Attribute Names
    //
    const string vsDataContainerId = "vsDataContainerId";
    const string vsDataType = "vsDataType";
    const string vsData = "vsData";
    const string vsDataFormatVersion = "vsDataFormatVersion";
};

//Defination for MO class InventoryUnit

interface InventoryUnit: Top
{
    const string CLASS = "InventoryUnit";

    //Attribute Names
    //
    const string inventoryUnitId = "inventoryUnitId";
```

```

const string inventoryUnitType = "inventoryUnitType";
const string vendorUnitFamilyType = "vendorUnitFamilyType";
const string vendorUnitTypeNumber = "vendorUnitTypeNumber";
const string vendorName = "vendorName";
const string serialNumber = "serialNumber";
const string versionNumber = "versionNumber";
const string dateOfManufacture = "dateOfManufacture";
const string dateOfLastService = "dateOfLastService";
const string unitPosition = "unitPosition";
const string manufacturerData = "manufacturerData";
};

```

//Definitions for MO class SignallingPoint

interface SignallingPoint: Top

```

{
    const string CLASS = "SignallingPoint";

    // Attribute Names
    //
    const string signallingPointId = "signallingPointId";
    const string signallingInfo = "signallingInfo";
    const string signallingPointType = "signallingPointType";
    const string userLabel = "userLabel";
};

```

interface SignallingLinkSetTP: Top

```

{
    const string CLASS = "SignallingLinkSetTP";

    // Attribute Names
    //
    const string signallingLinkSetTPId = "signallingLinkSetTPId";
    const string adjacentSignallingInfo = "adjacentSignallingInfo";
    const string userLabel = "userLabel";
    const string signallingLinkType = "signallingLinkType";
};

```

interface SignallingLinkTP: Top

```

{
    const string CLASS = "SignallingLinkTP";

    // Attribute Names
    //
    const string signallingLinkTPId = "signallingLinkTPId";
};

```



```

const string userLabel = "userLabel";
const string slc = "slc";
const string slsNormalList = "slsNormalList";
const string slsCurrentList = "slsCurrentList";
const string linkStatus = "linkStatus";
const string bandwidth = "bandwidth";

};

};

#endif // _GENERIC_NRM_DEFS_IDL_

```

4.1.2 GenericNRMPProfile

```

//File "GenericNRMPProfile.idl"
//The IRP document version number is "Generic NRM V1.1"
#ifndef _GENERIC_NRM_PROFILE_IDL_
#define _GENERIC_NRM_PROFILE_IDL_

#include "GenericNRMSystem.idl"

/**
 * This module defines the attribute names and
 * correspondig attribute types for all defined
 * MO class. This module is used for reference.
 */
module GenericNRMPProfile
{
    interface Top
    {
        readonly attribute string objectClass;
        readonly attribute string objectInstance;
    };

    interface IRPAgent : Top
    {
        readonly attribute GenericNRMSystem::ObjectIdType iRPAgentId;
        readonly attribute GenericNRMSystem::DN systemDN;

        // The following notifications may be sent from this MO,
        // notifyObjectCreation
        // notifyObjectDeletion
        // notifyAttributeValueChange
        // notifyAckStateChanged
        // notifyChangedAlarm
        // notifyClearedAlarm
    };
};

```

```

// notifyNewAlarm
// notifyComments
// notifyAlarmListRebuilt
// notifyPotentialFaultyAlarmList
};

interface GenericIRP : Top
{
    readonly attribute string iRPIId;
};

interface SubNetwork: Top
{
    readonly attribute GenericNRMSSystem::ObjectIdType subNetworkId;
    readonly attribute GenericNRMSSystem::DNPrefixType dnPrefix;
        attribute wstring userLabel;
    readonly attribute GenericNRMSSystem::MobileCountryCodeSetType setOfMcc;
    readonly attribute GenericNRMSSystem::NetworkTypeType userDefinedNetworkType;

    // The following notifications may be sent from this MO,
    // notifyObjectCreation
    // notifyObjectDeletion
    // notifyAttributeValueChange
    // notifyAckStateChanged
    // notifyChangedAlarm
    // notifyClearedAlarm
    // notifyNewAlarm
    // notifyComments
    // notifyAlarmListRebuilt
    // notifyPotentialFaultyAlarmList
};

interface MeContext: Top
{
    readonly attribute GenericNRMSSystem::ObjectIdType meContextId;
    readonly attribute GenericNRMSSystem::DNPrefixType dnPrefix;

    // The following notifications may be sent from this MO,
    // notifyObjectCreation
    // notifyObjectDeletion
    // notifyAttributeValueChange
};

```

```

interface ManagementNode : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType managementNodeId;
    readonly attribute GenericNRMSystem::DNListType managedElements;
        attribute wstring userLabel;
        attribute GenericNRMSystem::UserDefinedStateType userDefinedState;
    readonly attribute string swVersion; // software version
        attribute wstring locationName;
    readonly attribute string vendorName;

    // The following notifications may be sent from this MO,
        // notifyObjectCreation
        // notifyObjectDeletion
        // notifyAttributeValueChange
        // notifyAckStateChanged
        // notifyChangedAlarm
        // notifyClearedAlarm
        // notifyNewAlarm
        // notifyComments
        // notifyAlarmListRebuilt
        // notifyPotentialFaultyAlarmList
};

interface ManagedElement : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType managedElementId;
    readonly attribute GenericNRMSystem::DNPrefixType dnPrefix;
        attribute wstring userLabel;
    readonly attribute string vendorName;
        attribute wstring locationName;
    readonly attribute GenericNRMSystem::StringSet managedElementType;
    readonly attribute GenericNRMSystem::DN managedBy;
        attribute GenericNRMSystem::UserDefinedStateType userDefinedState;
    readonly attribute string swVersion; // software version

    // The following notifications may be sent from this MO,
        // notifyObjectCreation
        // notifyObjectDeletion
        // notifyAttributeValueChange
        // notifyAckStateChanged
        // notifyChangedAlarm
        // notifyClearedAlarm
        // notifyNewAlarm
        // notifyComments
        // notifyAlarmListRebuilt

```

```

// notifyPotentialFaultyAlarmList
};

interface ManagedFunction : Top
{
    attribute wstring userLabel;
};

interface VsDataContainer : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType vsDataContainerId;
    readonly attribute string vsDataType;
        attribute any vsData;
    readonly attribute string vsDataFormatVersion;

    // The following notifications may be sent from this MO,
        // notifyObjectCreation
        // notifyObjectDeletion
        // notifyAttributeValueChange
        // notifyAckStateChanged
        // notifyChangedAlarm
        // notifyClearedAlarm
        // notifyNewAlarm
        // notifyComments
        // notifyAlarmListRebuilt
        // notifyPotentialFaultyAlarmList
};

interface InventoryUnit : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType inventoryUnitId;
    readonly attribute string inventoryUnitType;
    readonly attribute string vendorUnitFamilyType;
    readonly attribute string vendorUnitTypeNumber;
    readonly attribute string vendorName;
    readonly attribute string serialNumber;
    readonly attribute string versionNumber;
    readonly attribute string dateOfManufacture;
    readonly attribute string dateOfLastService;
    readonly attribute wstring unitPosition;
    readonly attribute string manufacturerData;

    // The following notifications may be sent from this MO,
        // notifyObjectCreation
        // notifyObjectDeletion

```

```

// notifyAttributeValueChange
// notifyAckStateChanged
// notifyChangedAlarm
// notifyClearedAlarm
// notifyNewAlarm
// notifyComments
// notifyAlarmListRebuilt
// notifyPotentialFaultyAlarmList
};

interface SignallingPoint : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType signallingPointId;
    readonly attribute GenericNRMSystem::SignallingInfoType signallingInfo;
    readonly attribute GenericNRMSystem::SignallingPointType signallingPointType;
        attribute wstring userLabel;

    // The following notifications may be sent from this MO,
        // notifyObjectCreation
        // notifyObjectDeletion
        // notifyAttributeValueChange
        // notifyAckStateChanged
        // notifyChangedAlarm
        // notifyClearedAlarm
        // notifyNewAlarm
        // notifyComments
        // notifyAlarmListRebuilt
        // notifyPotentialFaultyAlarmList
};

interface SignallingLinkSetTP : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType signallingLinkSetTPId;
        attribute GenericNRMSystem::SignallingInfoType adjacentSignallingInfo;
        attribute wstring userLabel;
    readonly attribute GenericNRMSystem::SignallingLinkTypeType signallingLinkType;

    // The following notifications may be sent from this MO,
        // notifyObjectCreation
        // notifyObjectDeletion
        // notifyAttributeValueChange
        // notifyAckStateChanged
        // notifyChangedAlarm
        // notifyClearedAlarm
        // notifyNewAlarm

```

```

    // notifyComments
    // notifyAlarmListRebuilt
    // notifyPotentialFaultyAlarmList
};

interface SignallingLinkTP : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType signallingLinkTPId;
        attribute wstring userLabel;
    readonly attribute GenericNRMSystem::SlcType slc;
    readonly attribute GenericNRMSystem::SLSListType slsNormalList;
    readonly attribute GenericNRMSystem::SLSListType slsCurrentList;
    readonly attribute GenericNRMSystem::LinkStatusType linkStatus;
    readonly attribute GenericNRMSystem::BandwidthType bandwidth;

    // The following notifications may be sent from this MO,
    // notifyObjectCreation
    // notifyObjectDeletion
    // notifyAttributeValueChange
    // notifyAckStateChanged
    // notifyChangedAlarm
    // notifyClearedAlarm
    // notifyNewAlarm
    // notifyComments
    // notifyAlarmListRebuilt
    // notifyPotentialFaultyAlarmList
};
};
#endif // _GENERIC_NRM_PROFILE_IDL_

```

4.1.3 GenericNRMSystem

```

//File "GenericNRMSystem.idl"
//The IRP document version number is "Generic NRM V1.1"
#ifndef _GENERIC_NRM_SYSTEM_IDL_
#define _GENERIC_NRM_SYSTEM_IDL_

module GenericNRMSystem
{
    /**
     * This module adds datatype definitions for types
     * used in the Generic NRM which are not basic datatypes defined
     * already in CORBA.
     */
}

```

```

/**
 * The format of Distinguished Name (DN) is specified in "Name Conventions
 * for Managed Objects revision B".
 */
typedef string DN;

typedef sequence<DN> DNListType;

typedef string ObjectIDType;

typedef DN DNPrefixType;

typedef string MobileCountryCodeType;
typedef sequence<MobileCountryCodeType> MobileCountryCodeSetType;

typedef string NetworkTypeType;
const NetworkTypeType AN = "Access Netowrk";
const NetworkTypeType CN = "Core Netowrk";
const NetworkTypeType AN_CN = "AN and CN";

typedef unsigned long UserDefinedStateType;

/**
 * A set of strings.
 */
typedef sequence<string> StringSet;

enum NetworkIndicatorType
{
    international,
    spare,
    national,
    nationalSpare
};

enum SignallingPointLengthType
{
    bits_24,
    bits_14
};

struct SignallingInfoType
{
    SignallingPointLengthType signallingPointLength;
    unsigned long signallingPointCode;
}

```

```

    NetworkIndicatorType networkIndicator;
};

typedef unsigned long SignallingPointType;
const SignallingPointType SP=0;
const SignallingPointType HSTP=1;
const SignallingPointType LSTP=2;
const SignallingPointType HLSTP=3;

enum SignallingLinkTypeType
{
    N_SS7,
    W_SS7
};

typedef unsigned long linkDirectType;
const linkDirectType toHstp=1;
const linkDirectType toLstp=2;
const linkDirectType toCdmaGmsc=3;
const linkDirectType toMsc=4;
const linkDirectType toHlr=5;
const linkDirectType toMc=6;
const linkDirectType toScp=7;
const linkDirectType toBsc=8;
const linkDirectType toCncPstn=9;
const linkDirectType toCtPstn=10;
const linkDirectType toCmcc=11;
const linkDirectType toCtt=12;
const linkDirectType toVoiceMailBox=13;
const linkDirectType toColorRing=14;
const linkDirectType toGsm=15;
const linkDirectType toOthers=16;

typedef unsigned short SlcType;

typedef unsigned short SLSType;
typedef sequence<SLSType> SLSListType;

typedef unsigned short LinkStatusType;
const LinkStatusType normal_UDS = 0;
const LinkStatusType deactivated_UDS = 1;
const LinkStatusType failed_UDS = 2;
const LinkStatusType localBlocked_UDS = 3;
const LinkStatusType remoteBlocked_UDS = 4;
const LinkStatusType localInhibited_UDS = 5;

```



```

const LinkStatusType remoteInhibited_UDS = 6;

typedef unsigned long BandwidthType;

/**
 * Types for HRPD use
 */
typedef string IpAddressType;
typedef sequence<IpAddressType> IpAddressSet;
typedef string IpwithDomainType;
typedef sequence<IpwithDomainType> IpwithDomainSet;
typedef string IpwithMaskType;
typedef sequence<IpwithMaskType> IpwithMaskSet;
typedef string DomainwithIpAddrType;
typedef sequence<DomainwithIpAddrType> DomainwithIpAddrSet;
typedef string CodewithDomainType;
typedef sequence<CodewithDomainType> CodewithDomainSet;
};

#endif // _GENERIC_NRM_SYSTEM_IDL_

```

4.1.4 IMDataDefs

```

//File "IMDataDefs.idl"
//The IRP document version number is "Inventory NRM V1.0"
#ifndef _IM_DATA_DEFS_IDL_
#define _IM_DATA_DEFS_IDL_

#include "GenericNRMDefs.idl"

/**
 * This module defines constants for each MO class name and
 * the attribute names for each defined MO class.
 */
module IMDataDefs
{
    /**
     * Definitions for MO class InventoryUnit
     */
    interface InventoryUnit:GenericNRMDefs::Top
    {
        const string CLASS = "InventoryUnit";

        //Attribute Names
        //

```

```

const string inventoryUnitId = "inventoryUnitId";
const string inventoryUnitType = "inventoryUnitType";
const string vendorUnitFamilyType = "vendorUnitFamilyType";
const string vendorUnitTypeNumber = "vendorUnitTypeNumber";
const string vendorName = "vendorName";
const string serialNumber = "serialNumber";
const string versionNumber = "versionNumber";
const string dateOfManufacture = "dateOfManufacture";
const string dateOfLastService="dateOfLastService";
const string unitPosition = "unitPosition";
const string manufacturerData= "manufacturerData";

};

};

#endif // _IM_DATA_DEFS_IDL_

```

4.1.5 IMDataProfile

```

//File "IMDataProfile.idl"
//The IRP document version number is "Inventory NRM V1.0"
#ifndef _IM_DATA_PROFILE_IDL
#define _IM_DATA_PROFILE_IDL

#include "GenericNRMSystem.idl"
#include "GenericNRMProfile.idl"

module IMDataProfile
{
    interface InventoryUnit:GenericNRMProfile::Top
    {
        readonly attribute GenericNRMSystem::ObjectIdType inventoryUnitId;
        readonly attribute string inventoryUnitType ;
        readonly attribute string vendorUnitFamilyType;
        readonly attribute string vendorUnitTypeNumber;
        readonly attribute string vendorName;
        readonly attribute string serialNumber;
        readonly attribute string versionNumber;
        readonly attribute string dateOfManufacture;
        readonly attribute string dateOfLastService;
        readonly attribute wstring unitPosition;
        readonly attribute string manufacturerData;

        // The following notifications may be sent from this MO,
        // notifyObjectCreation
        // notifyObjectDeletion
    }
}

```

```

        // notifyAttributeValueChanged
        // notifyAckStateChanged
        // notifyChangedAlarm
        // notifyClearedAlarm
        // notifyNewAlarm
        // notifyComments
        // notifyAlarmListRebuilt
        // notifyPotentialFaultyAlarmList
    };
};

#endif // _IM_DATA_PROFILE_IDL

```

4.2 无线接入网网络资源模型的 IDL 定义

4.2.1 RanNRMDefs

```

//File RanNRMDefs.idl
//The IRP document version number is "Ran NRM V1.0"
#ifndef _RAN_NRM_DEFS_IDL_
#define _RAN_NRM_DEFS_IDL_

#include "GenericNRMDefs.idl"

/**
 * This module defines constants for each MO class name and
 * the attribute names for each defined MO class.
 */
module RanNRMDefs
{
    /**
     * Definitions for MO class HrpdBscFunction
     */
    interface HrpdBscFunction : GenericNRMDefs::ManagedFunction
    {
        const string CLASS = "HrpdBscFunction";

        //Attribute Names
        //
        const string hrpdBscFunctionId = "hrpdBscFunctionId";
        const string controlledBtsList = "controlledBtsList";
        const string relatedAnAaaAddrList = "relatedAnAaaAddrList";
        const string maxNumHrpdBscSession = "maxNumHrpdBscSession";
    };
    /**
     * Definitions for MO class BtsFunction

```

```

*/
interface BtsFunction : GenericNRMDefs::ManagedFunction
{
    const string CLASS = "BtsFunction";

    //Attribute Names
    //
    const string btsFunctionId = "btsFunctionId";
    const string relatedBsc = "relatedBsc";
    const string numFa = "numFa";
};

/**
 * Definetions for MO class HrpSector
 */
interface HrpSector : GenericNRMDefs::ManagedFunction
{
    const string CLASS = "HrpSector";

    //Attribute Names
    //
    const string hrpSectorId = "hrpSectorId";
    const string sectorAddrId = "sectorAddrId";
    const string subnetMask = "subnetMask";
    const string colorCode = "colorCode";
    const string relatedBts = "relatedBts";
};

/**
 * Definetions for MO class PcfFunction
 */
interface PcfFunction : GenericNRMDefs::ManagedFunction
{
    const string CLASS = "PcfFunction";

    //Attribute Names
    //
    const string pcfFunctionId = "pcfFunctionId";
    const string ipAddressList = "ipAddressList";
    const string relatedPdsnAddrList = "relatedPdsnAddrList";
    const string relatedBscList = "relatedBscList";
    const string throughputCapacity = "throughputCapacity";
    const string maxNumSupportedSession = "maxNumSupportedSession";
    const string maxNumSupportedActiveSession = "maxNumSupportedActiveSession";
};

/**

```

```

    * Definetions for MO class AnAaaFunction
    */
interface AnAaaFunction : GenericNRMDefs::ManagedFunction
{
    const string CLASS = "AnAaaFunction";

    //Attribute Names
    //
    const string anAaaFunctionId = "anAaaFunctionId";
    const string extIpAddress = "extIpAddress";
    const string maxNumUsers = "maxNumUsers";
    const string maxNumAuthProcessedPerSec = "maxNumAuthProcessedPerSec";
    const string authPort = "authPort";
    const string relatedAnAddrList = "relatedAnAddrList";
    const string radiusServerIpwithDomain = "radiusServerIpwithDomain";
    const string codeAnalysis = "codeAnalysis";
};

#endif // _RAN_NRM_DEFS_IDL_

```

4.2.2 RanNRMProfile

```

//File RanNRMProfile.idl
//The IRP document version number is "Ran NRM V1.0"
#ifndef _RAN_NRM_PROFILE_IDL_
#define _RAN_NRM_PROFILE_IDL_

#include "GenericNRMSystem.idl"
#include "GenericNRMProfile.idl"
#include "RanNRMSystem.idl"
/**
 * This module defines the attribute names and
 * corresponding attribute types for all defined
 * MO class in Ran network.This module is
 * used for reference.
 */
module RanNRMProfile
{
    interface HrpdBscFunction : GenericNRMProfile::ManagedFunction
    {
        readonly attribute GenericNRMSystem::ObjectIdType hrpdBscFunctionId;
        attribute GenericNRMSystem::DNListType controlledBtsList;
        readonly attribute GenericNRMSystem::IpAddressSet relatedAnAaaAddrList;
        readonly attribute unsigned long maxNumHrpdSession;
    }
}

```

```

// The following notifications may be sent from this MO,
// notifyObjectCreation
// notifyObjectDeletion
// notifyAttributeValueChange
// notifyAckStateChanged
// notifyChangedAlarm
// notifyClearedAlarm
// notifyNewAlarm
// notifyComments
// notifyAlarmListRebuilt
// notifyPotentialFaultyAlarmList
};
interface BtsFunction : GenericNRMPProfile::ManagedFunction
{
    readonly attribute GenericNRMSystem::ObjectIdType btsFunctionId;
    attribute GenericNRMSystem::DN relatedBsc;
    readonly attribute unsigned long numFa;

    // The following notifications may be sent from this MO,
    // notifyObjectCreation
    // notifyObjectDeletion
    // notifyAttributeValueChange
    // notifyAckStateChanged
    // notifyChangedAlarm
    // notifyClearedAlarm
    // notifyNewAlarm
    // notifyComments
    // notifyAlarmListRebuilt
    // notifyPotentialFaultyAlarmList
};
interface HrpdsSector : GenericNRMPProfile::ManagedFunction
{
    readonly attribute GenericNRMSystem::ObjectIdType hrpdSectorId;
    readonly attribute GenericNRMSystem::IpAddressType sectorAddrId;
    readonly attribute unsigned long subnetMask;
    readonly attribute unsigned long colorCode;
    readonly attribute unsigned long subnetMask;
    readonly attribute GenericNRMSystem::DN relatedBts;

    // The following notifications may be sent from this MO,
    // notifyObjectCreation
    // notifyObjectDeletion
    // notifyAttributeValueChange
    // notifyAckStateChanged

```

```

// notifyChangedAlarm
// notifyClearedAlarm
// notifyNewAlarm
// notifyComments
// notifyAlarmListRebuilt
// notifyPotentialFaultyAlarmList
};
interface PcfFunction : GenericNRMPProfile::ManagedFunction
{
    readonly attribute GenericNRMSystem::ObjectIdType pcfFunctionId;
    readonly attribute GenericNRMSystem::IpAddressSet ipAddressList;
    readonly attribute GenericNRMSystem::IpAddressSet relatedPdsnAddrList;
        attribute GenericNRMSystem::DNListType relatedBscList;
    readonly attribute unsigned long throughputCapacity;
    readonly attribute unsigned long maxNumSupportedSession;
    readonly attribute unsigned long maxNumSupportedActiveSession;

    // The following notifications may be sent from this MO,
    // notifyObjectCreation
    // notifyObjectDeletion
    // notifyAttributeValueChange
    // notifyAckStateChanged
    // notifyChangedAlarm
    // notifyClearedAlarm
    // notifyNewAlarm
    // notifyComments
    // notifyAlarmListRebuilt
    // notifyPotentialFaultyAlarmList
};
interface AnAaaFunction:GenericNRMPProfile::ManagedFunction
{
    readonly attribute GenericNRMSystem::ObjectIdType anAaaFunctionId;
    readonly attribute GenericNRMSystem::IpwithMaskSet extIpAddress;
    readonly attribute unsigned long maxNumUsers;
    readonly attribute unsigned long maxNumAuthProcessedPerSec;
    readonly attribute unsigned long authPort;
    readonly attribute GenericNRMSystem::IpwithMaskSet relatedAnAddrList;
    readonly attribute GenericNRMSystem::DomainwithIpAddrSet radiusServerIpwithDomain;
    readonly attribute GenericNRMSystem::CodewithDomainSet codeAnalysis;

    // The following notifications may be sent from this MO,
    // notifyObjectCreation
    // notifyObjectDeletion
    // notifyAttributeValueChange
    // notifyAckStateChanged

```

```

        // notifyChangedAlarm
        // notifyClearedAlarm
        // notifyNewAlarm
        // notifyComments
        // notifyAlarmListRebuilt
        // notifyPotentialFaultyAlarmList
    };
};

#endif // _RAN_NRM_PROFILE_IDL_

```

4.2.3 RanNRMSystem

```

//File RanNRMSystem.idl
//The IRP document version number is "Ran NRM V1.0"
#ifndef _RAN_NRM_SYSTEM_IDL_
#define _RAN_NRM_SYSTEM_IDL_

#include "GenericNRMSystem.idl"

/**
 * The following module 'RanNRMSystem' is noted for further extension
 */

// module RanNRMSystem
//{
//}

#endif // _RAN_NRM_SYSTEM_IDL_

```

4.3 核心网网络资源模型的 IDL 定义

4.3.1 CoreNRMDefs

```

//File: CoreNRMDefs.idl
//The IRP document version number is "Core NRM V1.0"
#ifndef _CORE_NRM_DEFS_IDL_
#define _CORE_NRM_DEFS_IDL_

#include "GenericNRMDefs.idl"

/**
 * This module defines constants for each MO class name and
 * the attribute name for each defined MO class.
 */
module CoreNRMDefs
{

```



```

/**
 * Definitions for MO class PdsnFunction
 */
interface PdsnFunction : GenericNRMDefs::ManagedFunction
{
    const string CLASS = "PdsnFunction";
    //Attribute Names
    //
    const string pdsnFunctionId = "pdsnFunctionId";
    const string intIpAddressList = "intIpAddressList";
    const string extIpAddressList = "extIpAddressList";
    const string careOfAddressList = "careOfAddressList";
    const string aaaClientIpAddrList = "aaaClientIpAddrList";
    const string relatedAaaAddrList = "relatedAaaAddrList";
    const string relatedPcfAddrList = "relatedPcfAddrList";
    const string lacIpAddrList = "lacIpAddrList";
    const string maxNumPppConns = "maxNumPppConns";
    const string maxNumPppConnsEstabedPerSec = "maxNumPppConnsEstabedPerSec";
    const string maxNumPcfs = "maxNumPcfs";
    const string authProtocol = "authProtocol";
    const string maxSipRpRegLifetime = "maxSipRpRegLifetime";
    const string maxMipRpRegLifetime = "maxMipRpRegLifetime";
    const string maxThroughput = "maxThroughput";
    const string maxL2tpHelloInterval = "maxL2tpHelloInterval";
};
/**
 * Definitions for MO class HaFunction
 */
interface HaFunction : GenericNRMDefs::ManagedFunction
{
    const string CLASS = "HaFunction";
    //Attribute Names
    //
    const string haFunctionId = "haFunctionId";
    const string intIpAddressList = "intIpAddressList";
    const string extIpAddressList = "extIpAddressList";
    const string maxNumSessions = "maxNumSessions";
    const string dataThroughput = "dataThroughput";
};
/**
 * Definitions for MO class AaaFunction
 */
interface AaaFunction : GenericNRMDefs::ManagedFunction
{
    const string CLASS = "AaaFunction";

```

```

//Attribute Names
//
const string aaaFunctionId = "aaaFunctionId";
const string extIpAddressList = "extIpAddressList";
const string maxNumUsers = "maxNumUsers";
const string maxNumAccAuthProcessedPerSec = "maxNumAccAuthProcessedPerSec";
const string authPort = "authPort";
const string acctPort = "acctPort";
const string relatedNeIpAddr = "relatedNeIpAddr";
const string codeAnalysis = "codeAnalysis";
const string radiusServerIpwithDomain = "radiusServerIpwithDomain";
const string multiIspRealm = "multiIspRealm";

};

};

#endif // _CORE_NRM_DEFS_IDL_

```

4.3.2 CoreNRMPProfile

```

//File CoreNRMPProfile.idl
//The IRP document version number is "Core NRM V1.0"
#ifndef _CORE_NRM_PROFILE_IDL_
#define _CORE_NRM_PROFILE_IDL_

#include "GenericNRMSystem.idl"
#include "GenericNRMPProfile.idl"
#include "CoreNRMSystem.idl"
/**
 * This module defines the attribute names and
 * corresponding attribute types for all defined
 * MO class in corenetwork.This module is
 * used for reference.
 */
module CoreNRMPProfile
{
    interface PdsnFunction : GenericNRMPProfile::ManagedFunction
    {
        readonly attribute GenericNRMSystem::ObjectIdType pdsnFunctionId;
        readonly attribute GenericNRMSystem::IpwithMaskSet intIpAddressList;
        readonly attribute GenericNRMSystem::IpwithMaskSet extIpAddressList;
        readonly attribute GenericNRMSystem::IpwithMaskSet careOfAddressList;
        readonly attribute GenericNRMSystem::IpwithMaskSet aaaClientIpAddrList;
        readonly attribute GenericNRMSystem::IpwithMaskSet relatedAaaAddrList;
        readonly attribute GenericNRMSystem::IpwithMaskSet relatedPcfAddrList;
        readonly attribute GenericNRMSystem::IpwithMaskSet lacIpAddrList;
    }
}

```

```

    readonly attribute unsigned long maxNumPppConns;
    readonly attribute unsigned long maxNumPppConnsEstabedPerSec;
    readonly attribute unsigned long maxNumPcfs;
    readonly attribute CoreNRMSystem::authProtocolType authProtocol;
    readonly attribute unsigned long maxSipRpRegLifetime;
    readonly attribute unsigned long maxMipRpRegLifetime;
    readonly attribute unsigned long maxThroughput;
    readonly attribute unsigned long maxL2tpHelloInterval;

    // The following notifications may be sent from this MO,
    // notifyObjectCreation
    // notifyObjectDeletion
    // notifyAttributeValueChange
    // notifyAckStateChanged
    // notifyChangedAlarm
    // notifyClearedAlarm
    // notifyNewAlarm
    // notifyComments
    // notifyAlarmListRebuilt
    // notifyPotentialFaultyAlarmList
};
interface HaFunction : GenericNRMPProfile::ManagedFunction
{
    readonly attribute GenericNRMSystem::ObjectIdType haFunctionId;
    readonly attribute GenericNRMSystem::IpwithMaskSet intIpAddressList;
    readonly attribute GenericNRMSystem::IpwithMaskSet extIpAddressList;
    readonly attribute unsigned long maxNumSessions;
    readonly attribute unsigned long dataThroughput;

    // The following notifications may be sent from this MO,
    // notifyObjectCreation
    // notifyObjectDeletion
    // notifyAttributeValueChange
    // notifyAckStateChanged
    // notifyChangedAlarm
    // notifyClearedAlarm
    // notifyNewAlarm
    // notifyComments
    // notifyAlarmListRebuilt
    // notifyPotentialFaultyAlarmList
};
interface AaaFunction : GenericNRMPProfile::ManagedFunction
{
    readonly attribute GenericNRMSystem::ObjectIdType aaaFunctionId;
    readonly attribute GenericNRMSystem::IpAddressSet extIpAddressList;

```

```

readonly attribute unsigned long maxNumUsers;
readonly attribute unsigned long maxNumAccAuthProcessedPerSec;
readonly attribute unsigned long authPort;
readonly attribute unsigned long acctPort;
readonly attribute GenericNRMSystem::IpAddressSet relatedNeIpAddr;
readonly attribute GenericNRMSystem::CodewithDomainSet codeAnalysis;
readonly attribute GenericNRMSystem::DomainwithIpAddrSet radiusServerIpwithDomain;
readonly attribute GenericNRMSystem::DomainwithIpAddrSet multiIspRealm;

// The following notifications may be sent from this MO,
// notifyObjectCreation
// notifyObjectDeletion
// notifyAttributeValueChange
// notifyAckStateChanged
// notifyChangedAlarm
// notifyClearedAlarm
// notifyNewAlarm
// notifyComments
// notifyAlarmListRebuilt
// notifyPotentialFaultyAlarmList
};
};
#endif // _CORE_NRM_PROFILE_IDL_

```

4.3.3 CoreNRMSystem

```

//File CoreNRMSystem.idl
//The IRP document version number is "Core NRM V1.0"
#ifndef _CORE_NRM_SYSTEM_IDL_
#define _CORE_NRM_SYSTEM_IDL_

#include "GenericNRMSystem.idl"

module CoreNRMSystem
{
    /**
     * This module adds datatype definitions for types
     * used in the CoreNRM which are not basic datatypes defined
     * already in CORBA and datatypes defined already in
     * GenericNRMSystem.
     */

    typedef unsigned short authProtocol;
    const authProtocol MSID = 0;

```

```

const authProtocol CHAP = 1;
const authProtocol PAP = 2;
typedef sequence<authProtocol> authProtocolType;

};

#endif // _CORE_NRM_SYSTEM_IDL_

```

5 性能网络资源模型设计

5.1 性能参数的 IDL 定义

5.1.1 HrpdMeasCollecDefs

```

//File HrpdMeasCollecDefs.idl

#ifndef _HRPD_MEAS_COLLEC_DEFS_IDL_
#define _HRPD_MEAS_COLLEC_DEFS_IDL_

/**
 * This module defines measurementType names constants
 */
module HrpdMeasCollecDefs
{
    /**
     * RAN part
     */

    //HrpdBscFunction measurement
    module anHrpdHoMeas
    {
        const string attIncHardHo = "attIncHardHo";
        const string succIncHardHo = "succIncHardHo";
        const string attSoftHoAddition = "attSoftHoAddition";
        const string succSoftHoAddition = "succSoftHoAddition";
        const string virtualHoAddition = "virtualHoAddition";
    };
    module anHrpdTrafFMeas
    {
        const string attPages = "attPages";
        const string succPages = "succPages";
        const string succPagesRes = "succPagesRes";
        const string nbrCallDrop = "nbrCallDrop";
        const string trafficByWalshFtch = "trafficByWalshFtch";
        const string trafficRtchCe = "trafficRtchCe";
        const string failCallsAnPart = "failCallsAnPart";
    };
};

```

```

const string trafficRxA8 = "trafficRxA8";
const string trafficTxA8 = "trafficTxA8";
const string nbrRlpTxByte = "nbrRlpTxByte";
const string nbrRlpReTxByte = "nbrRlpReTxByte";
const string nbrRlpRxByte = "nbrRlpRxByte";
const string nbrRlpRxReTxReqByte = "nbrRlpRxReTxReqByte";
const string nbrRlpRxMissingByte = "nbrRlpRxMissingByte";
};
module anHrpdSessionMeas
{
    const string attSessionAuth = "attSessionAuth";
    const string succSessionAuth = "succSessionAuth";
    const string attSessionSetup = "attSessionSetup";
    const string succSessionSetup = "succSessionSetup";
    const string attSessionNego = "attSessionNego";
    const string succSessionNego = "succSessionNego";
    const string nbrSessionNormRIs = "nbrSessionNormRIs";
    const string nbrSessionAbnRIs = "nbrSessionAbnRIs";
    const string maxHrpdSession = "maxHrpdSession";
    const string attSessionSubnetDormantHoAdd = "attSessionSubnetDormantHoAdd";
    const string succSessionSubnetDormantHoAdd = "succSessionSubnetDormantHoAdd";
    const string attSessionSubnetActiveHoAdd = "attSessionSubnetActiveHoAdd";
    const string succSessionSubnetActiveHoAdd = "succSessionSubnetActiveHoAdd";
    const string attSessionAnDormantHoAdd = "attSessionAnDormantHoAdd";
    const string succSessionAnDormantHoAdd = "succSessionAnDormantHoAdd";
};
//HrpdSector measurement
module sectorHrpdHoMeas
{
    const string attSoftHoAdd = "attSoftHoAdd";
    const string succSoftHoAdd = "succSoftHoAdd";
    const string virtualSoftHoAdd = "virtualSoftHoAdd";
};
module sectorHrpdTraffMeas
{
    const string attCalls = "attCalls";
    const string succCalls = "succCalls";
    const string attFastCalls = "attFastCalls";
    const string succFastCalls = "succFastCalls";
    const string nbrCallDrop = "nbrCallDrop";
    const string trafficMacFLPerRate = "trafficMacFLPerRate";
    const string trafficMacRLPerRate = "trafficMacRLPerRate";
};
//PCF measurement
module pcfMeas

```

```

{
    const string succAccessPs = "succAccessPs";
    const string failCallsPcfPart = "failCallsPcfPart";
    const string maxPppSessActive = "maxPppSessActive";
    const string maxPppSessDorm = "maxPppSessDorm";
    const string attCallsFromDormantToActive = "attCallsFromDormantToActive";
    const string succCallsFromDormantToActive = "succCallsFromDormantToActive";
    const string trafficRxA10 = "trafficRxA10";
    const string trafficTxA10 = "trafficTxA10";
};
//ANAAA measurement
module anAaaMeas
{
    const string radiusAuthServTotalAccessRequests = "radiusAuthServTotalAccessRequests";
    const string radiusAuthServTotalAccessAccepts = "radiusAuthServTotalAccessAccepts";
    const string radiusAuthServTotalAccessRejects = "radiusAuthServTotalAccessRejects";
    const string radiusAuthServTotalAccessRequestsDeniedPerCause =
"radiusAuthServTotalAccessRequestsDeniedPerCause";
    const string radiusAuthClientAccessRequests = "radiusAuthClientAccessRequests";
    const string radiusAuthClientAccessAccepts = "radiusAuthClientAccessAccepts";
    const string radiusAuthClientAccessRejects = "radiusAuthClientAccessRejects";
    const string radiusAuthClientTotalRetransRequestsFwToHomeRegion =
"radiusAuthClientTotalRetransRequestsFwToHomeRegion";
    const string RadiusAuthClientTotalInvalidResponsesFromHomeRegionPerCause =
"RadiusAuthClientTotalInvalidResponsesFromHomeRegionPerCause";
    const string numProcessAuthPerSec = "numProcessAuthPerSec";
};

/**
 * CN part
 */
//PDSN measurement
module pdsnMeas
{
    const string attInitA10Conn = "attInitA10Conn";
    const string succInitA10Conn = "succInitA10Conn";
    const string attRefA10Conn = "attRefA10Conn";
    const string succRefA10Conn = "succRefA10Conn";
    const string failA10ConnPerCause = "failA10ConnPerCause";
    const string attA10ConnRelbyPdsn = "attA10ConnRelbyPdsn";
    const string succA10ConnRelbyPdsn = "succA10ConnRelbyPdsn";
    const string attA10ConnRelByPcf = "attA10ConnRelByPcf";
    const string succA10ConnRelByPcf = "succA10ConnRelByPcf";
    const string failA10Session = "failA10Session";
    const string trafficSipRx = "trafficSipRx";
};

```

```

const string trafficSipTx = "trafficSipTx";
const string trafficMipRx = "trafficMipRx";
const string trafficMipTx = "trafficMipTx";
const string trafficL2tpRx = "trafficL2tpRx";
const string trafficL2tpTx = "trafficL2tpTx";
const string greStatsRx = "greStatsRx";
const string greStatsTx = "greStatsTx";
const string nbrPktsRpDrop = "nbrPktsRpDrop";
const string nbrPktsPiDrop = "nbrPktsPiDrop";
const string maxNbrPktsTransmittedPerSec = "maxNbrPktsTransmittedPerSec";
const string attPppConns = "attPppConns";
const string succPppConns = "succPppConns";
const string currPppSession = "currPppSession";
const string meanPppSessionsActive = "meanPppSessionsActive";
const string maxPppSessionsActive = "maxPppSessionsActive";
const string meanPppSession = "meanPppSession";
const string maxPppSession = "maxPppSession";
const string maxEstabPppConnsPerSec = "maxEstabPppConnsPerSec";
const string maxL2tpSessions = "maxL2tpSessions";
const string maxL2tpTunnels = "maxL2tpTunnels";
const string meanSessionTimePerAt = "meanSessionTimePerAt";
const string meanRxOctetsPerAt = "meanRxOctetsPerAt";
const string meanTxOctetsPerAt = "meanTxOctetsPerAt";
const string faRegReqReceived = "faRegReqReceived";
const string faRegRequestsRelayed = "faRegRequestsRelayed";
const string faRegReqDeniedPerCause = "faRegReqDeniedPerCause";
const string faRegRepliesRecieved = "faRegRepliesRecieved";
const string faRegRepliesRelayed = "faRegRepliesRelayed";
const string attL2tpTunnelEst = "attL2tpTunnelEst";
const string succL2tpTunnelEst = "succL2tpTunnelEst";
const string attL2tpSessionEst = "attL2tpSessionEst";
const string succL2tpSessionEst = "succL2tpSessionEst";
};

//HA measurement
module haMeas
{
    const string haServiceRequestsAccepted = "haServiceRequestsAccepted";
    const string haRegistrationAccepted = "haRegistrationAccepted";
    const string haMultiBindingUnsupported = "haMultiBindingUnsupported";
    const string haServiceRequestsDeniedPerCause = "haServiceRequestsDeniedPerCause";
    const string haRegRequestsReceived = "haRegRequestsReceived";
    const string haDeRegRequestsReceived = "haDeRegRequestsReceived";
    const string haRegRepliesSent = "haRegRepliesSent";
    const string haDeRegRepliesSent = "haDeRegRepliesSent";
}

```



```

};

//AAA measurement
module aaaMeas
{
    const string radiusAccServTotalRequests = "radiusAccServTotalRequests";
    const string radiusAccServTotalResponses = "radiusAccServTotalResponses";
    const string radiusAccServTotalRequestsDeniedPerCause =
"radiusAccServTotalRequestsDeniedPerCause";
    const string radiusAccClientTotalRequestsFwToServiceGw =
"radiusAccClientTotalRequestsFwToServiceGw";
    const string radiusAccClientTotalResponsesFwFromServiceGw =
"radiusAccClientTotalResponsesFwFromServiceGw";
    const string radiusAccClientTotalRetransRequestsFwToServiceGw =
"radiusAccClientTotalRetransRequestsFwToServiceGw";
    const string RadiusAccClientTotalInvalidResponsesFromServiceGwPerCause =
"RadiusAccClientTotalInvalidResponsesFromServiceGwPerCause";
    const string radiusAccClientTotalRequestsFwToHomeRegion =
"radiusAccClientTotalRequestsFwToHomeRegion";
    const string radiusAccClientTotalResponsesFwFromHomeRegion =
"radiusAccClientTotalResponsesFwFromHomeRegion";
    const string radiusAccClientTotalRetransRequestsFwToHomeRegion =
"radiusAccClientTotalRetransRequestsFwToHomeRegion";
    const string radiusAccClientTatalInvalidResponsesFromHomeRegionPerCause =
"radiusAccClientTatalInvalidResponsesFromHomeRegionPerCause";
    const string radiusAuthServTotalAccessRequests = "radiusAuthServTotalAccessRequests";
    const string radiusAuthServTotalAccessAccepts = "radiusAuthServTotalAccessAccepts";
    const string radiusAuthServTotalAccessRejects = "radiusAuthServTotalAccessRejects";
    const string radiusAuthServTotalAccessRequestsDeniedPerCause =
"radiusAuthServTotalAccessRequestsDeniedPerCause";
    const string radiusAuthClientAccessRequests = "radiusAuthClientAccessRequests";
    const string radiusAuthClientAccessAccepts = "radiusAuthClientAccessAccepts";
    const string radiusAuthClientAccessRejects = "radiusAuthClientAccessRejects";
    const string radiusAuthClientTotalRetransRequestsFwToHomeRegion =
"radiusAuthClientTotalRetransRequestsFwToHomeRegion";
    const string radiusAuthClientTotalInvalidResponsesFromHomeRegionPerCause =
"radiusAuthClientTotalInvalidResponsesFromHomeRegionPerCause";
    const string numProcessAccAuthPerSec = "numProcessAccAuthPerSec";
};
};

#endif // _HRPD_MEAS_COLLEC_DEFS_IDL_

```

5.2 数据类型的 IDL 定义

5.2.1 HrpdcMeasCollecSystem

```

//File HrpdcMeasCollecSystem.idl

#ifndef _HRPD_MEAS_COLLEC_SYSTEM_IDL_
#define _HRPD_MEAS_COLLEC_SYSTEM_IDL_

/**
 * This module defines type definitions for performance measurements
 */
module HrpdcMeasCollecSystem
{
    typedef unsigned long  CDMAMeasurementType1;
    typedef float  CDMAMeasurementType2;

    typedef unsigned long CauseType;
    const CauseType sum = 0;
    const CauseType other=65535;

/**
 * The following PerType/PerCause/PerRate definitions are defined for reference ONLY
 * because no precise Type/Cause/Rate sequence or id definition are restricted in 3GPP2/IETF
 * as done in 3GPP
 */

// The following A11SIG causes are defined for PDSN in the 3GPP2 A.S0008-0.
typedef CauseType A11SIGCause;

const A11SIGCause reason_unspecified_80H = 1;
const A11SIGCause administratively_prohibited_81H = 2;
const A11SIGCause insufficient_resources_82H = 3;
const A11SIGCause mobile_node_failed_authentication_83H = 4;
const A11SIGCause identification_mismatch_85H = 5;
const A11SIGCause poorly_formed_request_86H = 6;
const A11SIGCause unknown_PDSN_address_88H = 7;
const A11SIGCause requested_reverse_tunnel_unavailable_89H = 8;
const A11SIGCause T_bit_not_set_8AH = 9;
const A11SIGCause unsupported_vendor_ID_8DH = 10;

// The following MIP causes are defined for PDSN in the IETF RFC2006.
typedef CauseType MIPCause;

const MIPCause reason_unspecified = 64;
const MIPCause administratively_prohibited = 65;

```

```
const MIPCause insufficient_resources = 66;
const MIPCause mobile_node_failed_authentication = 67;
const MIPCause home_agent_failed_authentication = 68;
const MIPCause requested_lifetime_too_long = 69;
const MIPCause poorly_formed_request = 70;
const MIPCause poorly_formed_reply = 71;
const MIPCause requested_encapsulation_unavailable = 72;
const MIPCause requested_Van = 73;
const MIPCause home_agent_unreachable = 95;
//here the value '95' actually represents 80-95 in MIP Registration Relpy MIB
const MIPCause reasonUnspecified = 128;
const MIPCause admProhibited = 129;
const MIPCause insufficientResource = 130;
const MIPCause mnAuthenticationFailure = 131;
const MIPCause faAuthenticationFailure = 132;
const MIPCause idMismatch = 133;
const MIPCause poorlyFormedRequest = 134;
const MIPCause tooManyBindings = 135;
const MIPCause unknownHA = 136;
const MIPCause other_reason = 159;
//here the value '159' actually represents 137-159 in MIP Registration Relpy MIB

// The following RadiusAccServPerCause is defined for RADIUS in the IETF RFC2621/2866.
typedef CauseType RadiusAccServPerCause;

const RadiusAccServPerCause accServ_unknown_addresses = 1;
const RadiusAccServPerCause accServ_duplicate_requests = 2;
const RadiusAccServPerCause accServ_malformed_requests = 3;
const RadiusAccServPerCause accServ_invalid_signature_attributes = 4;
const RadiusAccServPerCause accServ_discarded_requests = 5;

// The following RadiusAccServPerCause is defined for RADIUS in the IETF RFC2620/2866.
typedef CauseType RadiusAccClientPerCause;

const RadiusAccClientPerCause accClient_unknown_addresses = 1;
const RadiusAccClientPerCause accClient_malformed_responses = 2;
const RadiusAccClientPerCause accClient_invalid_signature_attributes = 3;
const RadiusAccClientPerCause accClient_unknown_type = 4;
const RadiusAccClientPerCause accClient_discarded_packets = 5;

// The following RadiusAuthServPerCause is defined for RADIUS in the IETF RFC2619/2865.
typedef CauseType RadiusAuthServPerCause;

const RadiusAuthServPerCause authServ_unknown_addresses = 1;
const RadiusAuthServPerCause authServ_duplicated_requests = 2;
```

```

const RadiusAuthServPerCause authServ_input_error = 3;
const RadiusAuthServPerCause authServ_invalid_signature_attributes = 4;
const RadiusAuthServPerCause authServ_unknown_type_requests = 5;
const RadiusAuthServPerCause authServ_malformed_requests = 6;
const RadiusAuthServPerCause authServ_discarded_packets = 7;

// The following RadiusAuthClientPerCause is defined for RADIUS in the IETF RFC2618/2866.
typedef CauseType RadiusAuthClientPerCause;

const RadiusAuthClientPerCause authClient_unknown_addresses = 1;
const RadiusAuthClientPerCause authClient_malformed_responses = 2;
const RadiusAuthClientPerCause authClient_invalid_signature_attributes = 3;
const RadiusAuthClientPerCause authClient_unknown_type_requests = 4;
const RadiusAuthClientPerCause authClient_discarded_packets = 5;

typedef unsigned long RateType;

// The following trafficMacFLPerRate is defined for MAC_FL.
typedef RateType trafficMacFLPerRate;

const trafficMacFLPerRate fl_38.4_kbps = 1;
const trafficMacFLPerRate fl_76.8_kbps = 2;
const trafficMacFLPerRate fl_153.6_kbps = 3;
const trafficMacFLPerRate fl_S307.2_kbps = 4;
const trafficMacFLPerRate fl_L307.2_kbps = 5;
const trafficMacFLPerRate fl_S614.4_kbps = 6;
const trafficMacFLPerRate fl_L614.4_kbps = 7;
const trafficMacFLPerRate fl_912.6_kbps = 8;
const trafficMacFLPerRate fl_S1.228_mbps = 9;
const trafficMacFLPerRate fl_L1.228_mbps = 10;
const trafficMacFLPerRate fl_1.843_mbps = 11;
const trafficMacFLPerRate fl_2.457_mbps = 12;

// The following trafficMacRLPerRate is defined for MAC_RL.
typedef RateType trafficMacRLPerRate;

const trafficMacRLPerRate rl_9.6_kbps = 1;
const trafficMacRLPerRate rl_19.2_kbps = 2;
const trafficMacRLPerRate rl_38.4_kbps = 3;
const trafficMacRLPerRate rl_76.8_kbps = 4;
const trafficMacRLPerRate rl_153.6_kbps = 5;

};

#endif // _HRPD_MEAS_COLLEC_SYSTEM_IDL_

```

6 性能管理接口功能相关的文件

6.1 性能测量数据文件的 Schema 定义<HrpdMeasCollec.xsd>

下面的Schema文件中用到的字段的说明参见附录A，示例参见附录B。

版本号：1.0

```

<?xml version="1.0" encoding="UTF-8"?>
<!--REFERENCE: 3GPP TS 32.435 V7.2.0 measCollec.xsd-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:mc="std://yd-t/hrpd/itf-n/mc/2007"
targetNamespace="std://yd-t/hrpd/itf-n/mc/2007" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.0">
  <xsd:element name="measCollecFile">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="fileHeader">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="fileSender">
                <xsd:complexType>
                  <xsd:attribute name="localDn" type="xsd:string" use="optional"/>
                  <xsd:attribute name="elementType" type="xsd:string"
use="optional"/>
                  <xsd:attribute name="inDomain" type="mc:domainType"
use="optional"/>
                  <!--add an optional attribute "inDomain" indicating in which domain
the ME resides-->
                </xsd:complexType>
              </xsd:element>
            <xsd:element name="measCollec">
              <xsd:complexType>
                <xsd:attribute name="beginTime" type="xsd:dateTime"
use="required"/>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
          <xsd:attribute name="fileFormatVersion" type="xsd:string" use="required"/>
          <xsd:attribute name="vendorName" type="xsd:string" use="optional"/>
          <xsd:attribute name="dnPrefix" type="xsd:string" use="optional"/>
        </xsd:complexType>
      </xsd:element>
    <xsd:element name="measData" minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="managedElement">
            <xsd:complexType>

```

```

        <xsd:attribute name="localDn" type="xsd:string" use="optional"/>
        <xsd:attribute name="userLabel" type="xsd:string"
use="optional"/>
        <xsd:attribute name="swVersion" type="xsd:string"
use="optional"/>
        <xsd:attribute name="inDomain" type="mc:domainType"
use="optional"/>
        <!--add an optional attribute "inDomain" indicating in which domain
the ME resides-->
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="measInfo" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="job" minOccurs="0">
                    <xsd:complexType>
                        <xsd:attribute name="jobId" type="xsd:string"
use="required"/>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="granPeriod">
                    <xsd:complexType>
                        <xsd:attribute name="duration" type="xsd:duration"
use="required"/>
                        <xsd:attribute name="endTime"
type="xsd:dateTime" use="required"/>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="repPeriod" minOccurs="0">
                    <xsd:complexType>
                        <xsd:attribute name="duration" type="xsd:duration"
use="required"/>
                    </xsd:complexType>
                </xsd:element>
                <xsd:choice>
                    <xsd:element name="measTypes">
                        <xsd:simpleType>
                            <xsd:list itemType="mc:measName"/>
                            <!--restriction xsd:Name to mc:measName-->
                        </xsd:simpleType>
                    </xsd:element>
                    <xsd:element name="measType" minOccurs="0"
maxOccurs="unbounded">
                        <xsd:complexType>
                            <xsd:simpleContent>

```

```

type="xsd:positiveInteger" use="required"/>
                                <xsd:extension base="mc:measName">
                                    <xsd:attribute name="p"
type="xsd:positiveInteger" use="required"/>
                                </xsd:extension>
                                <!--restriction xsd:Name to
mc:measName-->
                                    </xsd:simpleContent>
                                    </xsd:complexType>
                                </xsd:element>
                                </xsd:choice>
                                <xsd:element name="measValue" minOccurs="0"
maxOccurs="unbounded">
                                    <xsd:complexType>
                                        <xsd:sequence>
                                            <xsd:choice>
                                                <xsd:element name="measResults">
                                                    <xsd:simpleType>
                                                        <xsd:list
itemType="mc:measResultType"/>
                                                    </xsd:simpleType>
                                                </xsd:element>
                                                <xsd:element name="r" minOccurs="0"
maxOccurs="unbounded">
                                                    <xsd:complexType>
                                                        <xsd:simpleContent>
                                                            <xsd:extension
base="mc:measResultType">
                                                                <xsd:attribute
name="p" type="xsd:positiveInteger" use="required"/>
                                                                </xsd:extension>
                                                            </xsd:simpleContent>
                                                        </xsd:complexType>
                                                    </xsd:element>
                                                </xsd:choice>
                                                <xsd:element name="suspect"
type="xsd:boolean" minOccurs="0"/>
                                            </xsd:sequence>
                                        <xsd:attribute name="measObjLdn"
type="xsd:string" use="required"/>
                                    </xsd:complexType>
                                </xsd:element>
                                </xsd:sequence>
                                <xsd:attribute name="measInfold" type="xsd:string"
use="optional"/>
                                </xsd:complexType>

```

```

        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="fileFooter">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="measCollec">
          <xsd:complexType>
            <xsd:attribute name="endTime" type="xsd:dateTime"
use="required"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:simpleType name="measResultType">
  <xsd:union memberTypes="xsd:decimal">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="NIL"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>
<!--New attribute type-->
<xsd:simpleType name="domainType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="CN_PS"/>
    <xsd:enumeration value="RAN"/>
    <xsd:enumeration value="IMS"/>
    <xsd:enumeration value="CN_CS"/>
    <xsd:enumeration value="WIN"/>
    <xsd:enumeration value="OTHER"/>
    <!--Legacy compatibility -->
  </xsd:restriction>
</xsd:simpleType>
<!--PM name extensions-->
<xsd:simpleType name="measName">
  <xsd:union memberTypes="mc:measNameWithSubCounter mc:measNameWithoutSubCounter"/>
</xsd:simpleType>
<xsd:simpleType name="measNameWithSubCounter">

```



```

    <xsd:restriction base="xsd:string">
      <xsd:pattern
value="(pdsnMeas.failA10ConnPerCause.|pdsnMeas.faRegReqDeniedPerCause.|lhaMeas.haServiceRequestsDenied
PerCause.|laaaMeas.radiusAccServTotalRequestsDeniedPerCause.|laaaMeas.RadiusAccClientTotalInvalidResponses
FromServiceGwPerCause.|laaaMeas.radiusAccClientTatalInvalidResponsFromHomeRegionPerCause.|laaaMeas.radi
usAuthServTotalAccessRequestsDeniedPerCause.|laaaMeas.radiusAuthClientTotalInvalidResponsesFromHomeReg
ionPerCause.|sectorHrpdTraffMeas.trafficMacFLPerRate.|sectorHrpdTraffMeas.trafficMacRLPerRate.|lanAaaMeas.
radiusAuthServTotalAccessRequestsDeniedPerCause.|lanAaaMeas.RadiusAuthClientTotalInvalidResponsesFromH
omeRegionPerCause.)\d{1,5}"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="measNameWithoutSubCounter">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="pdsnMeas.attInitA10Conn"/>
      <xsd:enumeration value="pdsnMeas.succInitA10Conn"/>
      <xsd:enumeration value="pdsnMeas.attRefA10Conn"/>
      <xsd:enumeration value="pdsnMeas.succRefA10Conn"/>
      <xsd:enumeration value="pdsnMeas.attA10ConnRelbyPdsn"/>
      <xsd:enumeration value="pdsnMeas.succA10ConnRelbyPdsn"/>
      <xsd:enumeration value="pdsnMeas.attA10ConnRelByPef"/>
      <xsd:enumeration value="pdsnMeas.succA10ConnRelByPcf"/>
      <xsd:enumeration value="pdsnMeas.failA10Session"/>
      <xsd:enumeration value="pdsnMeas.trafficSipRx"/>
      <xsd:enumeration value="pdsnMeas.trafficSipTx"/>
      <xsd:enumeration value="pdsnMeas.trafficMipRx"/>
      <xsd:enumeration value="pdsnMeas.trafficMipTx"/>
      <xsd:enumeration value="pdsnMeas.trafficL2tpRx"/>
      <xsd:enumeration value="pdsnMeas.trafficL2tpTx"/>
      <xsd:enumeration value="pdsnMeas.greStatsRx"/>
      <xsd:enumeration value="pdsnMeas.greStatsTx"/>
      <xsd:enumeration value="pdsnMeas.nbrPktsRpDrop"/>
      <xsd:enumeration value="pdsnMeas.nbrPktsPiDrop"/>
      <xsd:enumeration value="pdsnMeas.maxNbrPktsTransmittedPerSec"/>
      <xsd:enumeration value="pdsnMeas.attPppConns"/>
      <xsd:enumeration value="pdsnMeas.succPppConns"/>
      <xsd:enumeration value="pdsnMeas.currPppSession"/>
      <xsd:enumeration value="pdsnMeas.meanPppSessionsActive"/>
      <xsd:enumeration value="pdsnMeas.maxPppSessionsActive"/>
      <xsd:enumeration value="pdsnMeas.meanPppSession"/>
      <xsd:enumeration value="pdsnMeas.maxPppSession"/>
      <xsd:enumeration value="pdsnMeas.maxEstabPppConnsPerSec"/>
      <xsd:enumeration value="pdsnMeas.maxL2tpSessions"/>
      <xsd:enumeration value="pdsnMeas.maxL2tpTunnels"/>
      <xsd:enumeration value="pdsnMeas.meanSessionTimePerAt"/>
      <xsd:enumeration value="pdsnMeas.meanRxOctetsPerAt"/>
    </xsd:restriction>
  </xsd:simpleType>

```

```

<xsd:enumeration value="pdsnMeas.meanTxOctetsPerAt"/>
<xsd:enumeration value="pdsnMeas.faRegReqReceived"/>
<xsd:enumeration value="pdsnMeas.faRegRequestsRelayed"/>
<xsd:enumeration value="pdsnMeas.faRegRepliesRecieved"/>
<xsd:enumeration value="pdsnMeas.faRegRepliesRelayed"/>
<xsd:enumeration value="pdsnMeas.attL2tpTunnelEst"/>
<xsd:enumeration value="pdsnMeas.succL2tpTunnelEst"/>
<xsd:enumeration value="pdsnMeas.attL2tpSessionEst"/>
<xsd:enumeration value="pdsnMeas.succL2tpSessionEst"/>
<xsd:enumeration value="haMeas.haServiceRequestsAccepted"/>
<xsd:enumeration value="haMeas.haRegistrationAccepted"/>
<xsd:enumeration value="haMeas.haMultiBindingUnsupported"/>
<xsd:enumeration value="haMeas.haRegRequestsReceived"/>
<xsd:enumeration value="haMeas.haDeRegRequestsReceived"/>
<xsd:enumeration value="haMeas.haRegRepliesSent"/>
<xsd:enumeration value="haMeas.haDeRegRepliesSent"/>
<xsd:enumeration value="aaaMeas.radiusAccServTotalRequests"/>
<xsd:enumeration value="aaaMeas.radiusAccServTotalResponses"/>
<xsd:enumeration value="aaaMeas.radiusAccClientTotalRequestsFwToServiceGw"/>
<xsd:enumeration value="aaaMeas.radiusAccClientTotalResponsesFwFromServiceGw"/>
<xsd:enumeration value="aaaMeas.radiusAccClientTotalRetransRequestsFwToServiceGw"/>
<xsd:enumeration value="aaaMeas.radiusAccClientTotalRequestsFwToHomeRegion"/>
<xsd:enumeration value="aaaMeas.radiusAccClientTotalResponsesFwFromHomeRegion"/>
<xsd:enumeration value="aaaMeas.radiusAccClientTotalRetransRequestsFwToHomeRegion"/>
<xsd:enumeration value="aaaMeas.radiusAuthServTotalAccessRequests"/>
<xsd:enumeration value="aaaMeas.radiusAuthServTotalAccessAccepts"/>
<xsd:enumeration value="aaaMeas.radiusAuthServTotalAccessRejects"/>
<xsd:enumeration value="aaaMeas.radiusAuthClientAccessRequests"/>
<xsd:enumeration value="aaaMeas.radiusAuthClientAccessAccepts"/>
<xsd:enumeration value="aaaMeas.radiusAuthClientAccessRejects"/>
<xsd:enumeration value="aaaMeas.radiusAuthClientTotalRetransRequestsFwToHomeRegion"/>
<xsd:enumeration value="aaaMeas.numProcessAccAuthPerSec"/>
<xsd:enumeration value="anHrpdHoMeas.attIncHardHo"/>
<xsd:enumeration value="anHrpdHoMeas.succIncHardHo"/>
<xsd:enumeration value="anHrpdHoMeas.attSoftHoAddition"/>
<xsd:enumeration value="anHrpdHoMeas.succSoftHoAddition"/>
<xsd:enumeration value="anHrpdHoMeas.virtualHoAddition"/>
<xsd:enumeration value="anHrpdTraffMeas.attPages"/>
<xsd:enumeration value="anHrpdTraffMeas.succPages"/>
<xsd:enumeration value="anHrpdTraffMeas.succPagesRes"/>
<xsd:enumeration value="anHrpdTraffMeas.nbrCallDrop"/>
<xsd:enumeration value="anHrpdTraffMeas.trafficByWalshFtch"/>
<xsd:enumeration value="anHrpdTraffMeas.trafficRtchCe"/>
<xsd:enumeration value="anHrpdTraffMeas.failCallsAnPart"/>
<xsd:enumeration value="anHrpdTraffMeas.trafficRxAS8"/>

```

```

<xsd:enumeration value="anHrpdTraffMeas.trafficTxA8"/>
<xsd:enumeration value="anHrpdTraffMeas.nbrRlpTxByte"/>
<xsd:enumeration value="anHrpdTraffMeas.nbrRlpReTxByte"/>
<xsd:enumeration value="anHrpdTraffMeas.nbrRlpRxByte"/>
<xsd:enumeration value="anHrpdTraffMeas.nbrRlpRxReTxReqByte"/>
<xsd:enumeration value="anHrpdTraffMeas.nbrRlpRxMissingByte"/>
<xsd:enumeration value="anHrpdSessionMeas.attSessionAuth"/>
<xsd:enumeration value="anHrpdSessionMeas.succSessionAuth"/>
<xsd:enumeration value="anHrpdSessionMeas.attSessionSetup"/>
<xsd:enumeration value="anHrpdSessionMeas.succSessionSetup"/>
<xsd:enumeration value="anHrpdSessionMeas.attSessionNego"/>
<xsd:enumeration value="anHrpdSessionMeas.succSessionNego"/>
<xsd:enumeration value="anHrpdSessionMeas.nbrSessionNormRIs"/>
<xsd:enumeration value="anHrpdSessionMeas.nbrSessionAbnRIs"/>
<xsd:enumeration value="anHrpdSessionMeas.maxHrpdSession"/>
<xsd:enumeration value="anHrpdSessionMeas.attSessionSubnetDormantHoAdd"/>
<xsd:enumeration value="anHrpdSessionMeas.succSessionSubnetDormantHoAdd"/>
<xsd:enumeration value="anHrpdSessionMeas.attSessionSubnetActiveHoAdd"/>
<xsd:enumeration value="anHrpdSessionMeas.succSessionSubnetActiveHoAdd"/>
<xsd:enumeration value="anHrpdSessionMeas.attSessionAnDormantHoAdd"/>
<xsd:enumeration value="anHrpdSessionMeas.succSessionAnDormantHoAdd"/>
<xsd:enumeration value="sectorHrpdHoMeas.attSoftHoAdd"/>
<xsd:enumeration value="sectorHrpdHoMeas.succSoftHoAdd"/>
<xsd:enumeration value="sectorHrpdHoMeas.virtualSoftHoAdd"/>
<xsd:enumeration value="sectorHrpdTraffMeas.attCalls"/>
<xsd:enumeration value="sectorHrpdTraffMeas.succCalls"/>
<xsd:enumeration value="sectorHrpdTraffMeas.attFastCalls"/>
<xsd:enumeration value="sectorHrpdTraffMeas.succFastCalls"/>
<xsd:enumeration value="sectorHrpdTraffMeas.nbrCallDrop"/>
<xsd:enumeration value="pcfMeas.succAccessPs"/>
<xsd:enumeration value="pcfMeas.failCallsPcfPart"/>
<xsd:enumeration value="pcfMeas.maxPppSessActive"/>
<xsd:enumeration value="pcfMeas.maxPppSessDorm"/>
<xsd:enumeration value="pcfMeas.attCallsFromDormantToActive"/>
<xsd:enumeration value="pcfMeas.succCallsFromDormantToActive"/>
<xsd:enumeration value="pcfMeas.trafficRxA10"/>
<xsd:enumeration value="pcfMeas.trafficTxA10"/>
<xsd:enumeration value="anAaaMeas.radiusAuthServTotalAccessRequests"/>
<xsd:enumeration value="anAaaMeas.radiusAuthServTotalAccessAccepts"/>
<xsd:enumeration value="anAaaMeas.radiusAuthServTotalAccessRejects"/>
<xsd:enumeration value="anAaaMeas.radiusAuthClientAccessRequests"/>
<xsd:enumeration value="anAaaMeas.radiusAuthClientAccessAccepts"/>
<xsd:enumeration value="anAaaMeas.radiusAuthClientAccessRejects"/>
<xsd:enumeration value="anAaaMeas.radiusAuthClientTotalRetransRequestsFwToHomeRegion"/>
<xsd:enumeration value="anAaaMeas.numProcessAuthPerSec"/>

```

```
</xsd:restriction>  
</xsd:simpleType>  
</xsd:schema>
```

6.2 性能测量数据文件的 XML header 定义

在实际性能测量数据文件中应该使用下面的XML header定义:

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/xsl" href="HrpdMeasCollec.xsl"?>  
<measCollecFile xmlns="std://yd-t/hrpd/itf-n/mc/2007">
```

广东省网络空间安全协会受控资料

附录 A
(规范性附录)

XML Schema 文档补充说明

A.1 XML Schema文档结构标记

XML Schema文档结构标记约定如图A.1所示。

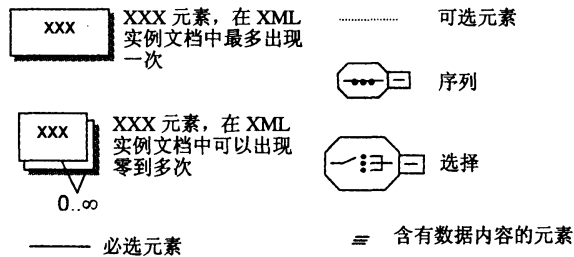


图 A.1 XML Schema 文档结构标记

A.2 性能测量数据文件的Schema定义

XML Schema文档<HrpdMeasCollec.xsd>结构如图A.2所示。

XML Schema文档元素/属性说明见表A.1。

表 A.1 HrpdMeasCollec.xsd 文档元素/属性说明

元素/属性名称		元素/属性描述
元素名称	包含属性	
measCollecFile		性能数据采集文件。是该Schema的根元素。由三个子元素组成：文件头部(fileHeader)、采集数据(measData)以及文件尾部(FileFooter)
fileHeader	fileFormatVersion	文件头部。由两个子元素组成：文件发送方(fileSender)、测量采集开始时间(measCollec)。包含三个属性：文件格式版本(fileFormatVersion)、制造商名称(vendorName)和识别名前缀(dnPrefix)
	vendorName	
	dnPrefix	
measData		性能测量数据。在一份采集上报文件中可出现零(未采集到数据)至多次。由两个子元素组成：管理网元(managedElement)及其性能采集结果(measInfo)
fileFooter		文件尾部。包含子元素：测量采集结束时间(measCollec)
fileSender	localDn	文件发送方。包含两个属性：本地识别名(localDN)、网元类型(elementType)
	elementType	
managedElement	localDn	被管网元。包括三个属性：本地识别名(localDn)、用户友好名(userLabel)、软件版本(swVersion)
	userLabel	
	swVersion	
measInfo	measInfoId	测量信息。由四个子元素组成：测量任务(job)、测量粒度周期(granPeriod)、测量上报周期(repPeriod)、测量类型(measType/measTypes)和测量值(measValue)。包含一个可选属性：测量信息标识符(measInfoId)
job	jobId	测量任务。该元素为可选元素。由其属性jobId唯一标识
granPeriod	duration	测量粒度周期。包含两个属性：持续时间(duration)、结束时间(endTime)
	endTime	
repPeriod	duration	测量上报周期。该元素为可选元素。包含唯一属性：持续时间(duration)
measTypes/measType		采集类型。均由measName扩展而来。在XML文件实例中，两个元素择一使用。不同的是measTypes是以列表方式呈现，且只出现一次；measType可出现多次，由属性值为非负数的p加以区分
measType	p	p为属性限定(position)。属性用于区分不同的measType

表A.1 (续)

元素/属性名称		元素/属性描述
元素名称	包含属性	
measResults/r		采集结果。均由measResultType扩展而来。在XML文件实例中，两个元素择一使用。值为空表示该采集项的取值无法获得。不同的是measResults是以列表方式呈现，且只出现一次；r可出现多次，由属性值为非负数p加以区分。r的p属性应与measType的p属性一一对应
r	p	p为属性限定。表示对<measType p>的一个采集结果应答。<r p>需和<measType p>一一对应
measValue	measObjLdn	采集值。由两个子元素组成：采集结果列表(measResults/r)和一个标记采集数据是否可信的标志位(suspect)。本身还包含一个属性：测量对象本地识别名(measObjLdn)
suspect		用于标记采集值是否可信。默认值为False(即可信)
measCollec	beginTime	性能采集开始时间
	endTime	性能采集结束时间
measName		性能测量项名称。分为包含子测量项(measNameWithSubCounter)和不含子测量项(measNameWithoutSubCounter)两类。从3GPP TS 32.435中扩展而来
measNameWithSubCounter		含子测量项的数据测量项名称。表示为familyname.measurename.subcounter形式。从3GPP TS 32.435中扩展而来
measNameWithoutSubCounter		不含子测量项的数据测量项名称。表示为familyname.measurename形式。从3GPP TS 32.435中扩展而来

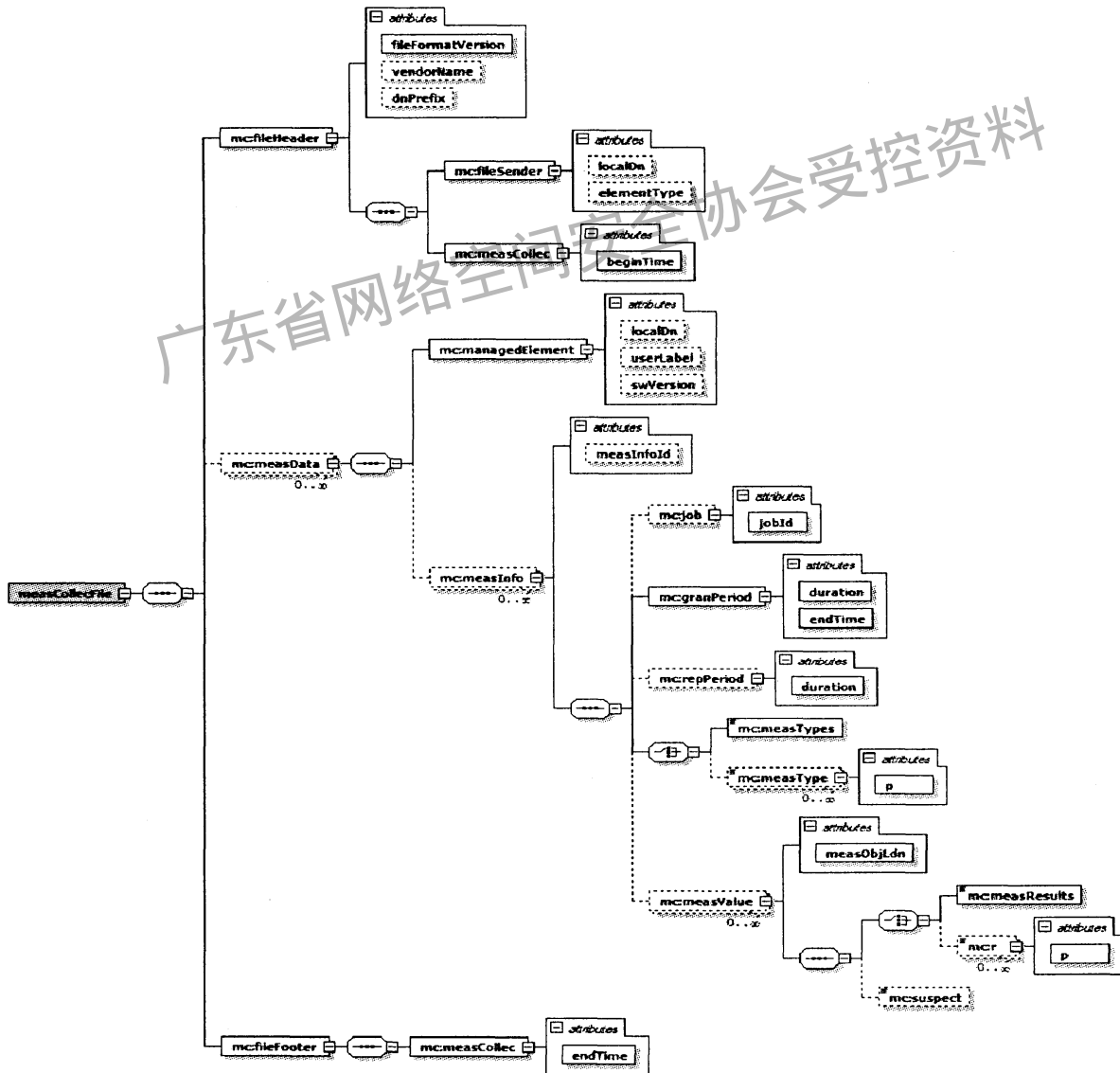


图 A.2 HrpDMeasCollec.xsd 文档结构

附录 B

(资料性附录)

性能管理功能相关 XML 文件示例

B.1 性能测量数据XML文件示例一：不使用可选的p属性

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="HrpdMeasCollec.xsl"?>
<!-- XML schema based XML measurement report file without use of optional positioning attributes on measurement types
and results. All values are hypothetical but syntactically correct -->
<measCollecFile xmlns="std://yd-t/hrpd/itf-n/mc/2007" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="std://yd-t/hrpd/itf-n/mc/2007 HrpdMeasCollec.xsd">
  <fileHeader fileFormatVersion="1.0" vendorName="Company NN" dnPrefix="DC=a1.companyNN.com,SubNetwork
=1,IRPAgent=1">
    <fileSender localDn="SubNetwork=CountryNN,MeContext=MEC-Gbg-1,ManagedElement=BSC-Gbg-1" element
Type="BSC" inDomain="RAN"/>
    <measCollec beginTime="2007-03-01T14:00:00+02:00"/>
  </fileHeader>
  <measData>
    <managedElement localDn="SubNetwork=CountryNN,MeContext=MEC-Gbg-1,ManagedElement=BSC-Gbg-1"
userLabel="BSC Telecomville"/>
    <measInfo>
      <job jobId="1231"/>
      <granPeriod duration="PT900S" endTime="2000-03-01T14:14:30+02:00"/>
      <repPeriod duration="PT1800S"/>
      <measTypes> mobileTrafficFlow.failOrigCallsPerCause.0
mobileTrafficFlow.failTermIncCallsPerCause.1 mobileTrafficFlow.failTermIncCallsPerCause.2 mobileTrafficFlow.failTerm
IncCallsPerCause.3 </measTypes>
      <measValue measObjLdn="BscFunction=RF-1,Sector=Gbg-997">
        <measResults>234 345 567 789</measResults>
      </measValue>
      <measValue measObjLdn="BscFunction=RF-1,Sector=Gbg-998">
        <measResults>890 901 123 234</measResults>
      </measValue>
      <measValue measObjLdn="BscFunction=RF-1,Sector=Gbg-999">
        <measResults>456 567 678 789</measResults>
        <suspect>true</suspect>
      </measValue>
    </measInfo>
  </measData>
</fileFooter>
  <measCollec endTime="2007-03-01T14:15:00+02:00"/>
</fileFooter>
</measCollecFile>

```

B.2 性能测量数据XML文件示例二：使用可选的p属性

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="HrpdMeasCollec.xsl"?>
<!-- XML schema based XML measurement report file with use of optional positioning attributes on measurement
types and results. All values are hypothetical but syntactically correct. -->
<measCollecFile xmlns="std://yd-t/hrpd/itf-n/mc/2007"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="std://yd-t/hrpd/itf-n/mc/2007
HrpdMeasCollec.xsd">
  <fileHeader fileFormatVersion="1.0" vendorName="Company NN"
dnPrefix="DC=a1.companyNN.com,SubNetwork=1,IRPAgent=1">
    <fileSender
localDn="SubNetwork=CountryNN,MeContext=MEC-Gbg-1,ManagedElement=BSC-Gbg-1"
elementType="BSC" inDomain="RAN"/>
      <measCollec beginTime="2007-03-01T14:00:00+02:00"/>
    </fileHeader>
    <measData>
      <managedElement
localDn="SubNetwork=CountryNN,MeContext=MEC-Gbg-1,ManagedElement=BSC-Gbg-1" userLabel="BSC
Telecomville"/>
        <measInfo>
          <job jobId="1231"/>
          <granPeriod duration="PT900S" endTime="2000-03-01T14:14:30+02:00"/>
          <repPeriod duration="PT1800S"/>
          <measType p="1">mobileTrafficFlow.failOrigCallsPerCause.0</measType>
          <measType p="2">mobileTrafficFlow.failOrigCallsPerCause.1</measType>
          <measType p="3">mobileTrafficFlow.failOrigCallsPerCause.2</measType>
          <measType p="4">mobileTrafficFlow.failTermIncCallsPerCause.3</measType>
          <measValue measObjLdn="BscFunction=RF-1,Sector=Gbg-997">
            <r p="1">234</r>
            <r p="2">345</r>
            <r p="3">567</r>
            <r p="4">789</r>
          </measValue>
          <measValue measObjLdn="BscFunction=RF-1,Sector=Gbg-998">
            <r p="1">890</r>
            <r p="2">901</r>
            <r p="3">123</r>
            <r p="4">234</r>
          </measValue>
          <measValue measObjLdn="BscFunction=RF-1,Sector=Gbg-999">
            <r p="1">456</r>
            <r p="2">567</r>
            <r p="3">678</r>
            <r p="4">789</r>
          </measValue>
        </measInfo>
      </managedElement>
    </measData>
  </fileSender>
</measCollecFile>

```



```

        <suspect>true</suspect>
      </measValue>
    </measInfo>
  </measData>
<fileFooter>
  <measCollec endTime="2007-03-01T14:15:00+02:00"/>
</fileFooter>
</measCollecFile>

```

B.3 性能测量数据XML文件示例三：使用可选的measInfoId属性

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="HrpdMeasCollec.xsl"?>
<!-- XML schema based XML measurement report file with use of optional measInfoId attribute. All values are
hypothetical but syntactically correct. -->
<measCollecFile xmlns="std://yd-t/hrpd/itf-n/mc/2007"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="std://yd-t/hrpd/itf-n/mc/2007
HrpdMeasCollec.xsd">
  <fileHeader fileFormatVersion="1.0" vendorName="Company NN" dnPrefix="SubNetwork=1">
    <fileSender localDn="OMC_PS=10" elementType="Element Manager" inDomain="EMS"/>
    <measCollec beginTime="2007-03-01T14:00:00+02:00"/>
  </fileHeader>
  <measData>
    <managedElement localDn="ManagedElement=PS_Core" userLabel="SGSN" swVersion="R30.1.5"/>
    <measInfo measInfoId="Category A">
      <job jobId="01"/>
      <granPeriod duration="PT900S" endTime="2007-03-01T14:15:00+02:00"/>
      <repPeriod duration="PT1800S"/>
      <measTypes>MM.AttGprsAttach MM.SuccGprsAttach MM. AbortedGprsAttach MM.AttIntraSgsn
RaUpdate</measTypes>
      <measValue measObjLdn="SgsnFunction=1">
        <measResults>10 20 30 40</measResults>
      </measValue>
    </measInfo>
    <measInfo measInfoId="Category B">
      <job jobId="02"/>
      <granPeriod duration="PT900S" endTime="2007-03-01T14:15:00+02:00"/>
      <repPeriod duration="PT1800S"/>
      <measTypes>MM.AttCombiAttach MM.SuccCombiAttach MM. MM.AbortedCombiAttachMM.Att
CombiDetachMs</measTypes>
      <measValue measObjLdn="SgsnFunction=2">
        <measResults>10 20 30 40</measResults>
      </measValue>
    </measInfo>
    <measInfo measInfoId="Category C">

```

```
<job jobId="03"/>
<granPeriod duration="PT1800S" endTime="2007-03-01T14:15:00+02:00"/>
<repPeriod duration="PT900S"/>
<measTypes>MM.AttPsPagingProclu MM.SuccPsPagingProclu</measTypes>
<measValue measObjLdn="SgsnFunction=3">
  <measResults>25 25</measResults>
</measValue>
</measInfo>
</measData>
<fileFooter>
  <measCollec endTime="2007-03-01T14:15:00+02:00"/>
</fileFooter>
</measCollecFile>
```

广东省网络空间安全协会受控资料

参 考 文 献

- [1] 3GPP TS 32.432 V7.0.0 性能测量—文件格式定义
 - [2] 3GPP TS 32.435 V7.2.0 性能测量—XML 文件格式定义
-

广东省网络空间安全协会受控资料

广东省网络空间安全协会受控资料

中华人民共和国
通信行业标准

800MHz/2GHz cdma2000 数字蜂窝移动通信网
高速分组数据（HRPD）网络管理技术要求
第3部分：基于 CORBA 技术的网络资源模型设计

YD/T 1878.3-2009

*

人民邮电出版社出版发行
北京市崇文区夕照寺街14号A座
邮政编码：100061

*

版权所有 不得翻印

*

本书如有印装质量问题，请与本社联系 电话：(010)67114922