CcspMoCA

- Introduction
- Code Flow
- Objects
- HAL APIs
- References

Introduction

This section summarises technical specifications for the operation of Multimedia Over Coax Alliance (MoCA) devices (“nodes”) using in-home coaxial wiring for the transport of multimedia content. It describes the MoCA node protocol stack and the physical network model.

The MoCA system network model creates a coax network that supports communications between a convergence layer in one MoCA node to the corresponding convergence layer in another MoCA node. The protocol stack of a MoCA node is shown in Figure 1-1. The MoCA specification does not include layers above the convergence layer.

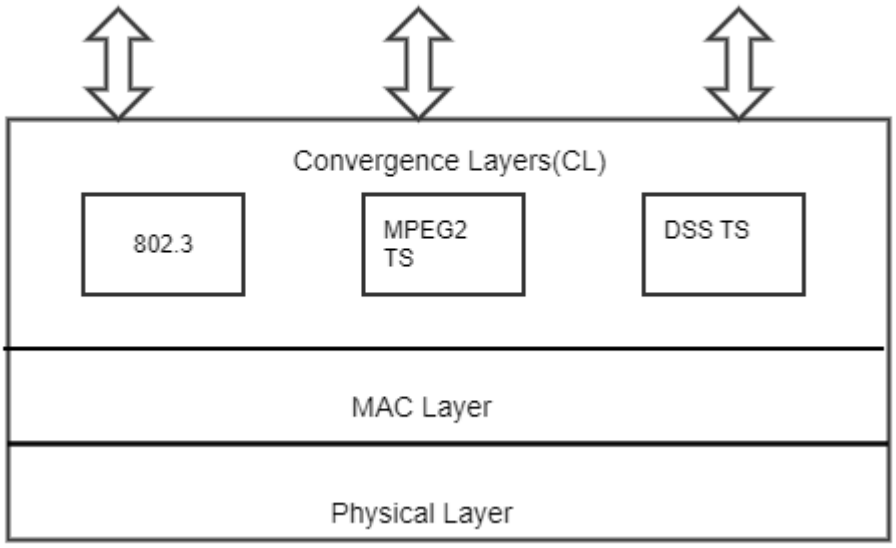

Figure 1-1: MoCA Node Protocol Stack

Physical Network Model

Typical in-home coaxial networks are configured as a branching tree topology. The point of connection to the first splitter is called the Root Node. The MoCA nodes inside the home communicate with each other by having their signals traverse across one or more splitters. The signal path transmission between two MoCA nodes is the superposition of several individual paths. Each individual signal path may have a different magnitude and delay resulting in an aggregate signal path with frequency nulls, large attenuation, and significant delay spread. The MoCA Network will operate under these channel conditions.

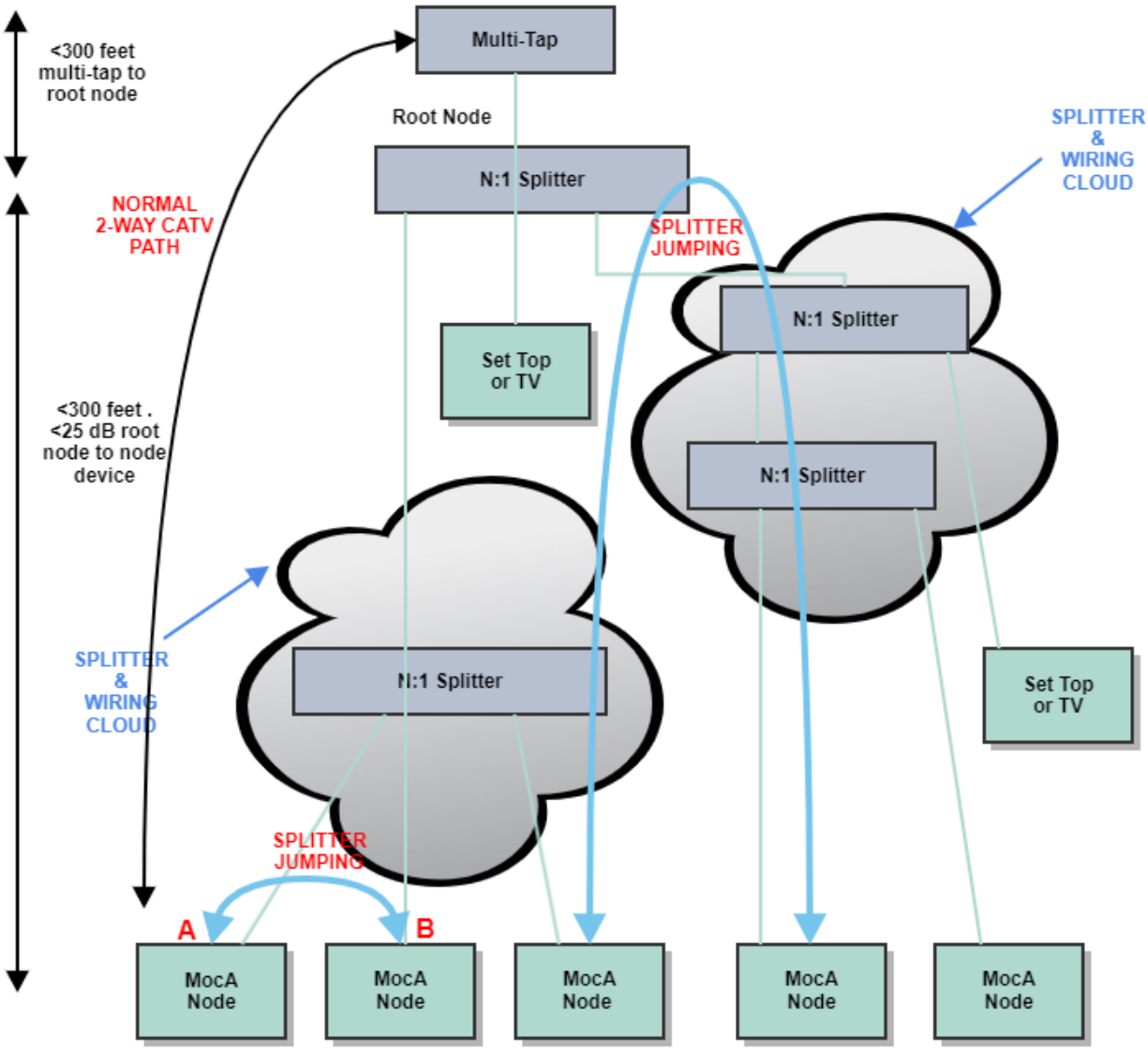

Figure 1-2: A Typical In-home MoCA Network

Code Flow

Moca Component registers with the CR using config files and XML files.

CcspMoCA.cfg specifies the datamodel, XmlConfig file CcspMoCADM.cfg which in turn specifies DmXml TR181-MoCA.XML

CcspMoCA.cfg :

Component>
<ID>com.cisco.spvtg.ccsp.moca</ID>
<Name>com.cisco.spvtg.ccsp.moca</Name>
<Version>1</Version>
<DbusPath>/com/cisco/spvtg/ccsp/moca</DbusPath>
<DataModelXmlCfg>CcspMoCADM.cfg</DataModelXmlCfg>
</Component>

CcspMoCADM.cfg:

<DataModels>
<DmXml>TR181-MoCA.XML</DmXml>
</DataModels>

- Name of the component =>com.cisco.spvtg.ccsp.moca
- Version of the component =>1

- D-Bus path to reach the component => /com/cisco/spvtg/ccsp/moca
- Namespaces supported by the component =>moca parameters
- If CcspMoca binary executes properly, a file /tmp/moca_initialized is created
- CcspMoca module consists of 2 parts
    - TR-181 - DataModel Layer provides details of parameters and objects supported by the moca component.
    - MoCASsp – involves Moca component registration with ComponentRegistry via CCsp DBUS

TR-181

Has two modules:-

- board_sbapi - Linked to moca-hal layer
- middle_layer_src – To handle query to moca parameters

Data model southbound API(board_sbapi) calls HAL APIs which abstract hardware configuration and status fetching.

The implementation of HAL APIs depends on platform SDK because different SoC vendors have different implementations for Moca.

When dmcli command is used to get or set a value, cosa_moca_dml.c in middle_layer_src is invoked.

```
For eg:  $ ./dmcli eRT getv "Device.MoCA.Interface.1.LinkUpTime"

               Component path is eRT.com.cisco.spvtg.ccsp.moca.
               Depending on parameter datatype (Bool/uLong/String) described in TR181-MoCA.XML, the
corresponding API in ./middle_layer_src/cosa_moca_dml.c
       (here, Interface1_GetParamUlongValue()) is called and collects values for the query parameter from moca
HAL layer through board-sbapi.
```

MocaSsp

- In **ssp_main.c, Cdm_Init(bus_handle, subSys, NULL, NULL, pComponentName);** Initializes Ccsp Datamodel for Mocawhere pComponentName =CCSP_COMPONENT_NAME_MoCA = com.cisco.spvtg.ccsp.moca" and subSys can be PC,Simulator,CableModem, eRouter etc

- **In ssp_messagebus_interface.c, ssp_Mbi_MessageBusEngage()** Connect to dbus daemon with component name, component path etc
- **In ssp_action.c,**
    - **Ssp_create()** Create component common data model object and interface ComponentCommonDmInit (g_pComponent_COMMON_moca);
    - **Ssp_engage()** Moca Data model configuration and registration
    - **RegisterCcspDataModel()** Register Moca DML with component name,path,version,namespace etc

  Once the system is fully initialized, set the health as Green

  g_pComponent_COMMON_moca->Health = CCSP_COMMON_COMPONENT_HEALTH_Green;

  Have some other functions like

    * ssp_cancel //to unregister component
    * ssp_CcdIfGetComponentName
    * ssp_CcdIfGetComponentVersion
    * ssp_CcdIfGetComponentAuthor

Objects

MoCA object in its DML layer:

```
Device.MoCA.Interface.
```

Retrieve value using dmcli command

```
$ dmcli eRT getv Device.MoCA.Interface.1.AssociatedDevice.

CR component name is: eRT.com.cisco.spvtg.ccsp.CR
subsystem_prefix eRT.
```

```
getv from/to component(eRT.com.cisco.spvtg.ccsp.moca): Device.MoCA.Interface.1.AssociatedDevice.
Execution succeed.
Parameter 1 name: Device.MoCA.Interface.1.AssociatedDevice.1.MACAddress
type: string, value: 12:BF:60:2E:4F:0D
Parameter 2 name: Device.MoCA.Interface.1.AssociatedDevice.1.NodeID
type: uint, value: 0
Parameter 3 name: Device.MoCA.Interface.1.AssociatedDevice.1.X_CISCO_COM_RxBcastRate
type: uint, value: 630
Parameter 4 name: Device.MoCA.Interface.1.AssociatedDevice.1.PreferredNC
type: bool, value: false
Parameter 5 name: Device.MoCA.Interface.1.AssociatedDevice.1.HighestVersion
type: string, value: 20
Parameter 6 name: Device.MoCA.Interface.1.AssociatedDevice.1.PHYTxRate
type: uint, value: 658
Parameter 7 name: Device.MoCA.Interface.1.AssociatedDevice.1.PHYRxRate
type: uint, value: 674
Parameter 8 name: Device.MoCA.Interface.1.AssociatedDevice.1.TxPowerControlReduction
type: uint, value: 30
Parameter 9 name: Device.MoCA.Interface.1.AssociatedDevice.1.RxPowerLevel
type: int, value: -47
Parameter 10 name: Device.MoCA.Interface.1.AssociatedDevice.1.TxBcastRate
type: uint, value: 641
Parameter 11 name: Device.MoCA.Interface.1.AssociatedDevice.1.RxBcastPowerLevel
type: uint, value: 0
Parameter 12 name: Device.MoCA.Interface.1.AssociatedDevice.1.TxPackets
type: uint, value: 5509
Parameter 13 name: Device.MoCA.Interface.1.AssociatedDevice.1.RxPackets
type: uint, value: 20680
Parameter 14 name: Device.MoCA.Interface.1.AssociatedDevice.1.RxErroredAndMissedPackets
type: uint, value: 0
Parameter 15 name: Device.MoCA.Interface.1.AssociatedDevice.1.QAM256Capable
type: bool, value: true
Parameter 16 name: Device.MoCA.Interface.1.AssociatedDevice.1.PacketAggregationCapability
type: uint, value: 1
Parameter 17 name: Device.MoCA.Interface.1.AssociatedDevice.1.RxSNR
type: uint, value: 8
Parameter 18 name: Device.MoCA.Interface.1.AssociatedDevice.1.Active
type: bool, value: true
Parameter 19 name: Device.MoCA.Interface.1.AssociatedDevice.1.X_CISCO_COM_NumberOfClients
type: uint, value: 0
Parameter 20 name: Device.MoCA.Interface.1.AssociatedDevice.2.MACAddress
type: string, value: D4:0A:A9:1B:5D:5C
Parameter 21 name: Device.MoCA.Interface.1.AssociatedDevice.2.NodeID
type: uint, value: 2
Parameter 22 name: Device.MoCA.Interface.1.AssociatedDevice.2.X_CISCO_COM_RxBcastRate
type: uint, value: 271
Parameter 23 name: Device.MoCA.Interface.1.AssociatedDevice.2.PreferredNC
type: bool, value: true
Parameter 24 name: Device.MoCA.Interface.1.AssociatedDevice.2.HighestVersion
type: string, value: 11
Parameter 25 name: Device.MoCA.Interface.1.AssociatedDevice.2.PHYTxRate
type: uint, value: 277
Parameter 26 name: Device.MoCA.Interface.1.AssociatedDevice.2.PHYRxRate
type: uint, value: 272
Parameter 27 name: Device.MoCA.Interface.1.AssociatedDevice.2.TxPowerControlReduction
type: uint, value: 29
Parameter 28 name: Device.MoCA.Interface.1.AssociatedDevice.2.RxPowerLevel
type: int, value: -53
Parameter 29 name: Device.MoCA.Interface.1.AssociatedDevice.2.TxBcastRate
type: uint, value: 274
Parameter 30 name: Device.MoCA.Interface.1.AssociatedDevice.2.RxBcastPowerLevel
type: uint, value: 0
Parameter 31 name: Device.MoCA.Interface.1.AssociatedDevice.2.TxPackets
type: uint, value: 4198
Parameter 32 name: Device.MoCA.Interface.1.AssociatedDevice.2.RxPackets
type: uint, value: 102237
Parameter 33 name: Device.MoCA.Interface.1.AssociatedDevice.2.RxErroredAndMissedPackets
type: uint, value: 0
Parameter 34 name: Device.MoCA.Interface.1.AssociatedDevice.2.QAM256Capable
type: bool, value: true
Parameter 35 name: Device.MoCA.Interface.1.AssociatedDevice.2.PacketAggregationCapability
```

```
type: uint, value: 1
Parameter 36 name: Device.MoCA.Interface.1.AssociatedDevice.2.RxSNR
type: uint, value: 8
Parameter 37 name: Device.MoCA.Interface.1.AssociatedDevice.2.Active
type: bool, value: true
Parameter 38 name: Device.MoCA.Interface.1.AssociatedDevice.2.X_CISCO_COM_NumberOfClients
type: uint, value: 0
Parameter 39 name: Device.MoCA.Interface.1.AssociatedDevice.3.MACAddress
type: string, value: 38:5F:66:16:2C:D1
Parameter 40 name: Device.MoCA.Interface.1.AssociatedDevice.3.NodeID
type: uint, value: 3
Parameter 41 name: Device.MoCA.Interface.1.AssociatedDevice.3.X_CISCO_COM_RxBcastRate
type: uint, value: 629
Parameter 42 name: Device.MoCA.Interface.1.AssociatedDevice.3.PreferredNC
type: bool, value: false
Parameter 43 name: Device.MoCA.Interface.1.AssociatedDevice.3.HighestVersion
type: string, value: 20
Parameter 44 name: Device.MoCA.Interface.1.AssociatedDevice.3.PHYTxRate
type: uint, value: 684
Parameter 45 name: Device.MoCA.Interface.1.AssociatedDevice.3.PHYRxRate
type: uint, value: 674
Parameter 46 name: Device.MoCA.Interface.1.AssociatedDevice.3.TxPowerControlReduction
type: uint, value: 30
Parameter 47 name: Device.MoCA.Interface.1.AssociatedDevice.3.RxPowerLevel
type: int, value: -50
Parameter 48 name: Device.MoCA.Interface.1.AssociatedDevice.3.TxBcastRate
type: uint, value: 641
Parameter 49 name: Device.MoCA.Interface.1.AssociatedDevice.3.RxBcastPowerLevel
type: uint, value: 0
Parameter 50 name: Device.MoCA.Interface.1.AssociatedDevice.3.TxPackets
type: uint, value: 2081
Parameter 51 name: Device.MoCA.Interface.1.AssociatedDevice.3.RxPackets
type: uint, value: 8589
Parameter 52 name: Device.MoCA.Interface.1.AssociatedDevice.3.RxErroredAndMissedPackets
type: uint, value: 0
Parameter 53 name: Device.MoCA.Interface.1.AssociatedDevice.3.QAM256Capable
type: bool, value: true
Parameter 54 name: Device.MoCA.Interface.1.AssociatedDevice.3.PacketAggregationCapability
type: uint, value: 1
Parameter 55 name: Device.MoCA.Interface.1.AssociatedDevice.3.RxSNR
type: uint, value: 8
Parameter 56 name: Device.MoCA.Interface.1.AssociatedDevice.3.Active
type: bool, value: true
Parameter 57 name: Device.MoCA.Interface.1.AssociatedDevice.3.X_CISCO_COM_NumberOfClients
type: uint, value: 0
```

HAL APIs

- MoCA HAL is an abstraction layer, implemented for interacting with MoCA driver.
- MoCA HAL API's functionality should be implemented by OEMs.
- *moca_hal.h* provides function call prototypes and structure definitions used for the RDK-Broadband MoCA HAL.
- Git repo: https://code.rdkcentral.com/r/plugins/gitiles/rdkb/components/opensource/ccsp/halinterface/+/rdk-next/moca_hal.h

Some example APIs are listed below:

APIs
moca_GetIfConfig
moca_SetIfConfig
moca_IfGetStaticInfo
moca_IfGetDynamicInfo
moca_IfGetStats
moca_GetNumAssociatedDevices
moca_GetAssociatedDevices

moca_GetResetCount
moca_GetFlowStatistics
moca_HardwareEquipped
moca_getIfScmod
moca_getIfAcaStatus
moca_cancelIfAca
moca_getIfAcaConfig
moca_setIfAcaConfig
moca_GetFullMeshRates
moca_FreqMaskToValue
moca_GetMocaCPEs
moca_IfGetExtAggrCounter
moca_IfGetExtCounter

References

- MoCA 1.1 Reference Specification
- MoCA Presentation