Overview

CoAP Simple Management Protocol (CSMP) is a device lifecycle management protocol optimized for resource constrained devices deployed within large-scale, bandwidth constrained IoT networks.

There are multiple target platforms supported by using OSAL (Operating System Abstraction Layer). The repository provides Linux, FreeRTOS and Silicon Labs EFR32 Wi-SUN device support. FreeRTOS initialised as a submodule and it is built with POSIX port. You can find details about Silicon Labs EFR32 Wi-SUN technology and products on silabs.com.

These instructions describe the build/run process for a C implementation of a sample CSMP Agent which incorporates the Cisco CSMP library `csmp-agent-lib`.

Building The CSMP Agent Sample and CSMP Agent Library

These instructions have been verified to work on Ubuntu 20.10 Desktop 64bit (RPi4 build and target platform). FreeRTOS platform uses port of POSIX OS API.

It is assumed a snapshot of the CSMP agent's source repository has been acquired and placed on the build platform via Github repository https://github.com/CiscoDevNet/csmp-agent-lib.

Change your default directory to the root of the CSMP Agent directory structure.

Install build tools.

Install build-essential package (gcc compiler, make, etc.) as described here ... https://help.ubuntu.com/community/InstallingCompilers

Confirm build target platform

If you are going to build for a different target platform, please set the correct gcc compiler for your target platform by modifying the line "CC=gcc" in "Makefile".

Build

chmod 777 build.sh

Linux

./build.sh linux

FreeRTOS

git submodule update –init –recursive ./build.sh freertos

If everything goes well, you should see “CsmpAgentLib_sample” executable in “sample” directory.

Silicon Labs - Wi-SUN EFR32 Platforms Silicon Labs Wi-SUN EFR32 platforms supports the CSMP Agent with the Simplicity SDK version 2024.6.0.

The demonstration requires using a Silicon Labs Wi-SUN Linux Border Router plus the Silicon Labs Wi-SUN node that will connect to Cisco FND.

Demonstration setup and detailed configuration can be found under the /Vendors/Silabs/ folder.

Clean

If you want to clean the build files prior to a subsequent build ... > ./build.sh clean

Debug output

Additional debug output is enabled by modifying Makefile to include the line ‘CFLAGS += -DPRINTDEBUG’

Running CSMP Agent Sample

1. Run “CsmpAgentLib_sample” to start CSMP agent either with: > ./CsmpAgentLib_sample -d

Or provide full comamnd line parameter set to configure FND server’s IPv6 address, agent’s registration interval (in seconds), EUI of the Agent (example) ... >./CsmpAgentLib_sample -d 2020::2020 -min 10 -max 100 -eid 00173B1122334455

NOTE: a valid FND IPv6 address must be supplied.

2. Once “csmpsagent” is started, it will begin registration attempts with the FND server.

Decoding CSMP Agent Messaging with Wireshark

Wireshark network analyzer may be used to observe CSMP messaging exchanged between the CSMP Agent and the FND instance. Note that this is a partial decode of the CoAP messaging and does not yet include decode of the TLV message payloads.

Install Wireshark

Follow the instructions here ... https://itsfoss.com/install-wireshark-ubuntu/. As of this writting, version 3.2.7 is installed.

Configure Wireshark for CSMP decoding.

Wireshark `Menu` -> `Analyze` -> `Decode As...` `+` -> `Field : UDP port` -> `value : 61628` -> `Current : CoAP` -> `OK`

Sample CSMP PCAP files.

Test your Wireshark install by opening and observing the contents of the sample PCAPs provided in the folder `test/*.pcap`.

TLV Support

CSMP messaging implements RESTful idioms with payloads encoded as Type/Length/Value tuples Value is encoded using Google Protocol Buffers. The Protocol Buffer definitions of CSMP TLVs are contained in the .proto file located in the src/csmpagent/tlvs folder. See 'src/csmpagent/csmpagent.c' for TLVs supported by the agent GET and POST methods.

Additions to the TLV set require ... 1. modification of the .proto file TLV definitions 2. compilation of the .proto file into .c and .h TLV files 3. rebuild of the agent (to use the new TLV files).

Use `protoc-c` (1.3.3 or later) to compile *.proto file into* .c and *.h files used during the agent build.

Install protoc-c

sudo apt-install protobuf-c-compiler

Go to src/csmpagent/tlvs/ and `make` to verify protoc-c is operating successfully.

Add TLVs

1. Assign new TLV ID XXX in 'src/csmpagent/csmp.h'
2. Add new TLV definition in `src/csmpagent/tlvs/CsmpTlvs.proto` and make to generate new CsmpTlvs.pb-c.c/CsmpTlvs.pb-c.h

Modify sample agent

1. Add desired GET or POST method dispatch for the new TLV XXX within 'src/csmpagent/csmpagent.c'.

2. Add required GET or POST implementations following the examples in folder 'src/csmpagent/'.

Further Information for Developers

A CSMP Developer Guide can be found in the /docs folder. This guide describes how to install, build, and run the CSMP agent which will register and report

metrics to an instance of Cisco Field Network Director.