CC31xx SLS Sniffer with Filters Application

Return to CC31xx & CC32xx Home Page

Return to CC31xx Sample Applications

Overview

This sample application demonstrates how NWP filters can be added to CC3100 for managing the Rx filtering process.

The Rx-Filters feature enables the user to simply define, and manage the Rx-filtering process. It reduces the amount of traffic transferred to the host, and achieves an efficient power management.

The host-driver provides a set of APIs to enable filter creation\editing. The APIs include the following operations:

- Add nodes: The user may request to add new node to the database, the node is created and a node unique ID is returned back to the user. The user should disable all filters before adding a node.
- Remove nodes: The user may request to remove a node from the filters' database,the node is removed. The user should disable all filters before removing a node.
- Enable nodes: The user can request to enable one or more filters, filters which are enabled, take part in the matching process.
- Disable nodes: The user can request to disable one or more filters, filters which are disabled, don take part in the matching process.
- Get filter node
- Update filter node

The application is responsible for adding filter nodes and defining the filters' tree hierarchy. If the user is interested in storing a filter node in the flash memory, he should set the persistence flag of the node. A filter can be defined as persistent only if its parent is persistent. Persistent filter nodes will be loaded automatically upon system startup.

The application developer can choose one or more filters to disable\enable, filters which are disabled don't take part in the matching process.

Note: This wiki page is only applicable for **CC3100-SDK v1.0.0** and upward releases. For documentation on older SDKs' examples, refer corresponding file in **<cc3100-sdk-installation-location>\cc3100-sdk\docs\examples\**

Application details

This application opens a raw socket in transceiver-mode, creates and applies filters based on the user input and prints the wireless data packets on the console

Filter can be created based on:

- Source MAC address
- Destination MAC address
- BSSID
- Frame type
- Frame Subtype (frame subtype filter requires a frame type filter as its parent): Some possible subtype bytes are
    - Probe Request: 40
    - Probe Response: 50
    - Beacon: 80
    - QOS Data: 88
    - Acknowledgement: d4
- Source IP address

- Destination IP address
- Packet length

Filter examples:

- To receive packets whose source MAC address is 'A' and destination MAC address is 'B':

1. Create a root filter **if (Source mac address != A), then Drop**
2. Create a leaf filter to the above parent filter **if (destination mac address != B), then Drop**

- To drop all the packets that are not probe-requests:

1. Create a root filter **if (frameType != management), then Drop**
2. Create a root filter **if (frameType == management), then NULL**
3. Create a leaf filter to parent filter #2 **if (frameSubType != probeRequest), then Drop**

For information on how to use Visual-Studio or Eclipse to compile and run this application, refer to cc3100_getting_started_guide_swru375 [1] in '<cc3100/>docs' folder.

By default, this application communicates w/ CC3100 over SPI. The SDK has UART-Drivers as well for 'SimpleLink Studio' platform. For using the UART interface to communicate w/ CC3100, macro **SL_IF_TYPE_UART** has to be defined in the application-project's properties. Also, 'COMM_PORT_NUM in **main.c** needs to be changed to the first com-port that gets enumerated for 'J6' of 'CC31xxEMUBOOST Brd'. In case four ports are getting enumerated, user should use the third com-port.

Usage

- Compile and run the application.
- Enter the channel and number of packets to be received.
- Enter 'f' to configure the filter.
- Select appropriate filter-parameters that are listed on the console to configure the required filters.
- Once the filter is configured, enter '10' to enable the filter and start receiving the packets.

Limitations/Known Issues

- Payload rule is currently not supported
- Filter's action of sending EVENT to the host upon a match is currently not supported.

References

[1] http://www.ti.com/lit/pdf/swru375

Article Sources and Contributors

CC31xx SLS Sniffer with Filters Application *Source*: http://processors.wiki.ti.com/index.php?oldid=185088 *Contributors*: A0131814, A0132173, A0221015, Codycooke, Malokyle

Image Sources, Licenses and Contributors

File:Cc31xx cc32xx return home.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Cc31xx_cc32xx_return_home.png *License*: unknown *Contributors*: A0221015

File:Cc31xx return sample apps.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Cc31xx_return_sample_apps.png *License*: unknown *Contributors*: A0221015