

Wireless Wi-Fi

Wi-Fi PNM Common Collection Framework Technical Report

WR-TR-PNM-WCCF-V01-171010

RELEASED

Notice

This Wi-Fi technical report is the result of a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry and its customers. You may download, copy, distribute, and reference the documents herein only for the purpose of developing products or services in accordance with such documents, and educational use. Except as granted by CableLabs in a separate written license agreement, no license is granted to modify the documents herein (except via the Engineering Change process), or to use, copy, modify or distribute the documents for any other purpose.

This document may contain references to other documents not owned or controlled by CableLabs. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in this document. To the extent this document contains or refers to documents of third parties, you agree to abide by the terms of any licenses associated with such third-party documents, including open source licenses, if any

DISCLAIMER

This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein. Any use or reliance on the information or opinion in this document is at the risk of the user, and CableLabs and its members shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, or utility of any information or opinion contained in the document.

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various entities, technology advances, or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein.

This document is not to be construed to suggest that any company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any of its members to purchase any product whether or not it meets the characteristics described in the document. Unless granted in a separate written agreement from CableLabs, nothing contained herein shall be construed to confer any license or right to any intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	WR-TR-PNM-WCCF-V01-171005			
Document Title:	Wi-Fi PNM Common Collection Framework Technical Report			
Revision History:	D01 - 02/28/17 D02 - 04/20/17 D03 - 05/16/17 V01 - 10/10/17			
Date:	October 10, 2017			
Status:	Work in Progress	Draft	Released	Closed
Distribution Restrictions:	Author Only	CL/Member	CL/Member/Vendor	Public

Trademarks

CableLabs® is a registered trademark of Cable Television Laboratories, Inc. Other CableLabs marks are listed at <http://www.cablelabs.com/certqual/trademarks>. All other marks are the property of their respective owners.

Contents

1	SCOPE.....	7
1.1	Introduction and Purpose.....	7
2	INFORMATIVE REFERENCES	8
2.1	Reference Acquisition.....	8
3	ABBREVIATIONS AND ACRONYMS.....	9
4	PROACTIVE NETWORK MAINTENANCE COMMON COLLECTION FRAMEWORK (CCF)	11
4.1	Wi-Fi PNM Common Collection Framework (WCCF).....	11
4.2	WCCF Modules.....	12
4.2.1	WCCF Agent.....	12
4.2.2	WCCF Collector	13
4.2.3	WCCF Controller	14
4.2.4	Demo Dashboard.....	14
5	WCCF SENSOR METRICS	15
5.1	WCCF Derived Values	21
5.1.1	STA Signal to Noise Ratio (SNR) and RSS.....	28
5.1.2	Channel Utilization, Channel Selection, and WLAN Traffic Metrics	28
5.1.3	MCS Index for HT and VHT	30
5.1.4	Frequency to Channel Conversion	33
5.1.5	Station and Channel Capabilities	36
APPENDIX I	LINUX WIRELESS NL80211 DRIVER API.....	38
I.1	General Attributes.....	39
I.2	Survey Collection Metrics	39
I.3	Station Collection Metrics	40
I.4	Scan Collection Metrics.....	41
I.5	Interfaces Collection Metrics.....	41
I.6	PhyCapa Collection Metrics	42
I.7	Event Collection Metrics	42
I.8	Process Collection Metrics	42
APPENDIX II	ACKNOWLEDGEMENTS	43

Figures

Figure 1: WCCF Architecture	11
Figure 2: WCCF Agent Sensor Examples	12
Figure 3: Example Dashboard for STA Statistics.....	14
Figure 4: Example Dashboard for Statistics for the AP of Interest.....	15
Figure 5: Example Dashboard of In-use and Available Channel Statistics	15
Figure 6: MCS Index and Data Rates	32
Figure 7: MCS Value Achieved by Clients at Various Signal to Noise Ratio (SNR)	33
Figure 8: nl80211 Interface	38

Tables

Table 1: WCCF Agent Modules	12
Table 2: Initial Wi-Fi Sensors.....	13
Table 3: Wi-Fi KPI metrics	16
Table 4: WCCF Data Elements	21
Table 5: nl80211_chan_width Enumeration Mapping.....	30
Table 6: 2.4 GHz Center Frequency (MHz) to Channel Lookup.....	34
Table 7: 5 GHz Center Frequency (MHz) to Channel Lookup.....	34
Table 8: Station Capability Classification	36
Table 9: General Attributes	39
Table 10: Survey Metrics used in KPI Calculation	39
Table 11: STA Metrics used in KPI Calculation	40
Table 12: Scan Metrics used in KPI Calculation	41
Table 13: Interface Metrics used in KPI Calculation.....	41
Table 14: PHY Capabilities Metrics used in KPI Calculation.....	42
Table 15: Process Metrics used in KPI Calculation.....	42

This page intentionally left blank

1 SCOPE

1.1 Introduction and Purpose

This document describes the Wi-Fi PNM Common Collection Framework architecture. Wi-Fi is no longer just a novelty or a product add-on; it's now an essential service. Customers rely heavily on Wi-Fi, and they know what bad Wi-Fi looks like – according to reports, a high percentage of support calls are Wi-Fi related. The need for Wi-Fi Proactive Network Maintenance (PNM) is overwhelming, and the Key Performance Indicators (KPI) defined in this document will establish the fundamental baseline required to create effective PNM delivery systems.

Data – both a lack of and an overabundance – is holding up efforts to enact Wi-Fi PNM. Cable operators have traditionally taken an all-or-nothing approach to polling information from a residential gateway, in that all MIBs or other data objects (if they are defined at all) are polled at once. Because there could be 400+ objects available, operators minimize the number of times a device is contacted, perhaps as infrequently as once a day. Wi-Fi is a dynamic technology and rapid response is critical, so the best way to deal with this infrequent polling is to focus the data collection on a few critical pieces of information. Delivery of this critical information through a framework that supports scheduled and asynchronous sending of only the critical information supports scalability. Coincidentally (and fortunately) the clear majority of all Wi-Fi support issues can be addressed by a handful of KPIs and an efficient framework where data can be analyzed at high resolution, but only distributed to OSS systems at lower resolution, or only upon an exceptional condition occurring in the home network.

Wi-Fi troubleshooting doesn't happen by looking at one single data point, so it's important to collectively evaluate all data. This release plan addresses a solution for scalable and efficient Wi-Fi PNM data collection with options for event-driven data distribution.

Additionally, CableLabs would like to facilitate better use of the DOCSIS® 3.1 PNM capabilities described in [PHYv3.1], [CM-OSSI], [CCAP-OSSIv3.1] and the architecture document for DOCSIS 3.1 Proactive Network Maintenance (PNM) Common Collection Framework (DCCF) [DCCFA]. The WCCF and DCCF will evolve towards a single PNM Common Collection Framework (CCF) supporting all CableLabs PNM efforts.

The architecture documentation for the overall PNM Common Collection Framework (CCF) that includes support for DOCSIS 3.1 PNM functions, the WCCF, and MOCA in the future is described in the DOCSIS 3.1 PNM Common Collection Framework Architecture Document [DCCFA]. Please refer to [DCCFA] to understand how the WCCF is a part of the vision for the overall CCF.

This document describes an aspirational architecture that will evolve in its development over time to support a field trial with a small number of CableLabs' members. Not all features, metrics, data elements, or nl80211 attributes may be included in the initial or following software releases. Each software release will prioritize the backlog set of both Wi-Fi metrics and KPIs along with architecture features. These priorities will be set by validation of the metrics and their effectiveness in mitigating Wi-Fi challenges. Refer to the CableLabs Wi-Fi PNM Common Collection Framework Requirements and Release Plan [WCCFRP] for scope and specific feature release planning. While best efforts will be made to keep this documentation in synch with WCCF software releases, if differences in data element names or units or formats, or architecture descriptions in this document exist reconciled with the WCCF code or [WCCFSW], the WCCF code and repository is the primary source of exact information on WCCF operation.

2 INFORMATIVE REFERENCES

This technical report uses the following informative references. References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific. For a non-specific reference, the latest version applies.

- [CM-OSSI] Cable Modem Operations Support System Interface Specification, CM-SP-CM-OSSIV3.1-I10-170906, September 6, 2017, CableLabs Television Laboratories.
- [CCAP-OSSIV3.1] DOCSIS 3.1 CCAP™ OSSI Specification, CM-SP-CCAP-OSSIV3.1-I10-170906, September 6, 2017, CableLabs Television Laboratories.
- [DCCFA] DOCSIS 3.1 Common Collection Framework (DCCF) Architecture Technical Report, CM-TR-DCCF-PNM-V01-171010, October 10, 2017, CableLabs Television Laboratories.
- [nl80211] Linux Wireless Wi-Fi Attributes kernel document, <http://git.kernel.org/cgit/linux/kernel/git/linville/wireless.git/tree/include/uapi/linux/nl80211.h?id=HEAD>
- [LW] Linux Wireless Wiki, <https://wireless.wiki.kernel.org/>
- [PHYv3.1] Physical Layer Specification, CM-SP-PHYv3.1-I11-170510, May 10, 2017, CableLabs Television Laboratories.
- [WCCFRP] CableLabs Wi-Fi PNM Common Collection Framework Requirements and Release Plan, to be released at a later date.
- [WCCFSW] WCCF Software Deployment, Administration and Release Notes along with documentation in the source tree with markdown and code comments, <https://code.cablelabs.com/CCF/wccf/tree/master/doc>
- [WKPI] [Top Key Performance Indicators in Wi-Fi](#), Technical Brief, July 2016.
- [WIFI-MGMT] Wi-Fi Provisioning Framework Specification, WR-SP-WiFi-MGMT-I08-161213, December 13, 2016, CableLabs Television Laboratories.
- [WIFI-GW] Wi-Fi Requirements for Cable Modem Gateways, WR-SP-WiFi-GW-I05-150515, May 15, 2015, CableLabs Television Laboratories.
- [WFTPG] Wi-Fi Field Trial Participant Guide, to be released at a later date.

2.1 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone +1-303-661-9100; Fax +1-303-661-9199; <http://www.cablelabs.com>
- Internet Engineering Task Force (IETF) Secretariat, 46000 Center Oak Plaza, Sterling, VA 20166, Phone +1-571-434-3500, Fax +1-571-434-3535, <http://www.ietf.org>

3 ABBREVIATIONS AND ACRONYMS

This document uses the following abbreviations:

ACK	Acknowledgement
ACS	Automatic Channel Selection
AP	Access Point
API	Application Programming Interface
BSS	Basic Service Set
C3	CableLabs Common Code Community
CCA	Clear Channel Assessment
CCAP	Converged Cable Access Platform. Often used interchangeably with CMTS, CCAP connotes a CMTS that also includes video capabilities
CCF	Common Collection Framework
CM	Cable Modem
CMTS	Cable Modem Termination System, often used interchangeably with CCAP
CS	Carrier Sense
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear to Send
DCCF	DOCSIS 3.1 Common Collection Framework
DOCSIS	Data Over Cable Service Interface Specification
DPoE	DOCSIS Provisioning of EPON
DPoG	DOCSIS Provisioning of GPON
ED	Energy Detection
EWMA	Exponential Weighted Moving Average
GI	Guard Interval
HT	High Throughput (802.11n channels)
IFS	Inter-frame Space
JSON	JavaScript Object Notation
KPI	Key Performance Indicators
MCS	Modulation and Coding Scheme
MIMO	Multiple Input Multiple Output
MoCA	Multimedia over Coax Alliance
MPDU	MAC Protocol Data Unit (contains a transmission the duration includes IFSs and ACK and potentially RTS CTS messages during a TXOP)
MVP	Minimum Viable Product
NAV	Network Allocation Vector
NB	Northbound
NF	Noise Figure
NSS	Number of Spatial Streams
OCD	OFDM Channel Description
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access

ONU	Optical Network Unit
OP	Observation Point
OSS	Operational Support Systems
OSSI	Operational Support Systems Interface
PDU	Protocol Data Unit
PHY	Physical Layer of the OSI model
PLCP	Physical Layer Convergence Protocol
PM	Performance Management
PMA	Profile Management Application
PNM	Proactive Network Maintenance
PPDU	PLCP Protocol Data Unit
QoE	Quality of Experience
REST	Representational State Transfer
RRM	Radio Resource Management
RSSI	Receive Signal Strength Indicator
RX	Receiver
RTS	Request To Send
S32I	Signed 32-bit Integer
SDN	Software Defined Networking
SID	Service ID
SNR	Signal to Noise Ratio
STA	Station
TX	Transmitter
TXOP	Transmission Opportunity
U32I	Unsigned 32-bit Integer
U64I	Unsigned 64-bit Integer
U8I	Unsigned 8-bit Integer
US	Upstream
VHT	Very High Throughput (802.11 ac channels)
VNF	Virtual Network Functions
WCCF	Wi-Fi Common Collection Framework
WLAN	Wireless Local Area Network
YANG	Yet Another Next Generation

4 PROACTIVE NETWORK MAINTENANCE COMMON COLLECTION FRAMEWORK (CCF)

The architecture documentation for the overall PNM Common Collection Framework (CCF) that includes support for DOCSIS 3.1 PNM functions, the WCCF, and possibly MOCA in the future, is described in the DOCSIS 3.1 PNM Common Collection Framework Architecture Document [DCCFA]. Please refer to [DCCFA] to understand how the WCCF is a part of the vision for the overall CCF.

4.1 Wi-Fi PNM Common Collection Framework (WCCF)

The overall WCCF is summarized in Figure 1, demonstrating the different components of a WCCF solution.

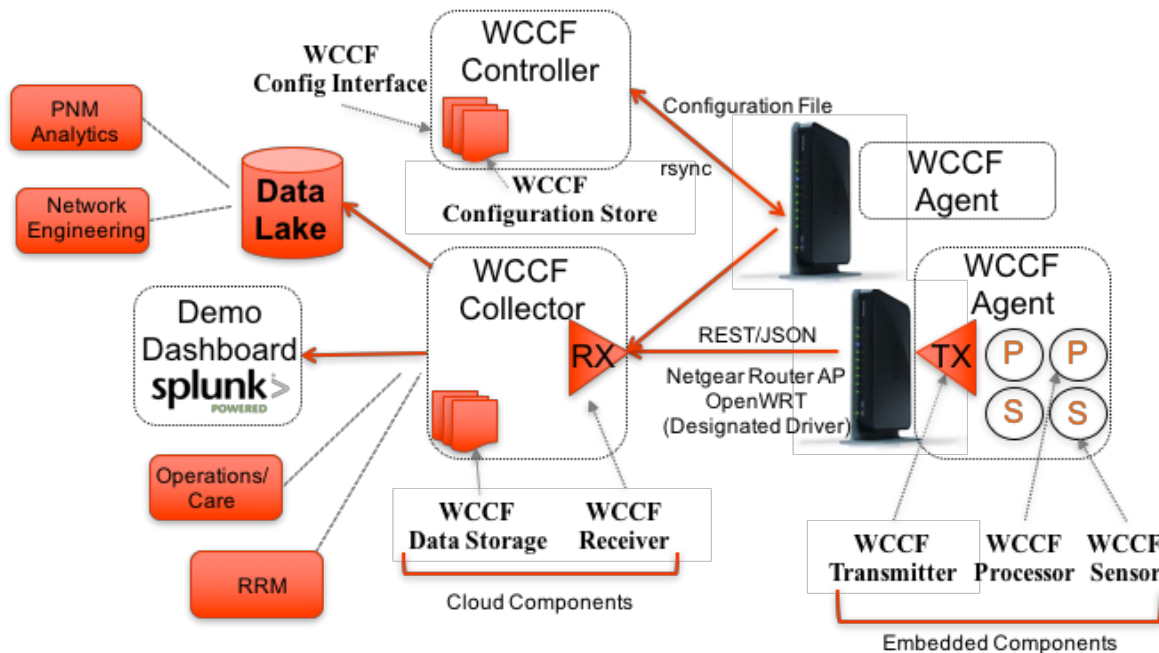


Figure 1: WCCF Architecture

The key architecture elements are summarized in this section, with more detail in the following subsections. The WCCF Agent is installed on a DOCSIS or PON gateway device with embedded AP, or a stand-alone AP behind the DOCSIS CM, or DPoE/DPoG ONU, or access network gateway. The WCCF agent has different software modules including a transmitter (TX) that sends data to the WCCF collector receiver (RX), WCCF sensors that obtain data from the device where the agent is installed (Wi-Fi AP), and WCCF processors, which can analyze data from the different sensors and manage workflow for transmission to the WCCF collector. The WCCF collector is where the data is stored and made available for a variety of applications such as PNM analytical applications, Radio Resource Management (RRM), Network Engineering, or technical support tools, among others. The WCCF keeps a short-term cache of data that can be leveraged for real time management and operations and can be synched with a larger data lake and enriched with other operational data for tending and analytics such as PNM, or network planning.

A new architectural element added is WCCF v1.4.0, called the Controller. This is a first MVP interface for setting some wireless and WCCF application configuration remotely over the network. The initial version is not a full configuration management framework, but provides basic support for configuration file updates that are synchronized to and from the AP via the controller using Linux rsync. More detailed descriptions of these modules are contained in the following sections and in the software documentation [WCCFSW].

4.2 WCCF Modules

The following sections describe the modules of the WCCF and their functions.

4.2.1 WCCF Agent

The WCCF agent includes three primary functional modules described in Table 1.

Table 1: WCCF Agent Modules

Modules	Description
Transmitter	Modular component that provides an interface for the transport protocol between the WCCF agent and the Receiver in the WCCF collector. Initially based on REST and JSON, the TX could support other protocols and data models in the future as alternatives such as TR232 and AMQP, along with many others. This document describes the data definitions; see detailed software documentation [WCCFSW] for the latest JSON description and examples.
Sensors	Sensors evaluate the SoC to obtain data from the embedded system. Initial sensors are focused on AP infrastructure Wi-Fi BSS, but in the future, could also include Wi-Fi Mesh, basic router and interface statistics, processor and environmental statistics, MOCA performance statistics, or other home network technology metrics. Only the sensors relevant to the device where the WCCF Agent resides and task will be deployed with the agent.
Processors	Processors take data from sensors, and based on policy or algorithms, may derive new metrics and make decisions on forwarding of data to the Transmitter.
Sync	Configuration files that set the APs wireless configuration including SSID, PSK, Channel, TX Power among other items, and WCCF software elements such as the collector and controller IP address, can be set on the controller and sent to the AP. A remote operation script can also be developed and supported by this interface to perform an operation such as remote software upgrade

4.2.1.1 WCCF Sensor

The WCCF agent is installed on each Wi-Fi AP that will be monitored. The transmitter and receiver functionality as described in Figure 2 allows for many sensors to send data to the WCCF collector, and enables redundant connections from the home network to the collector.

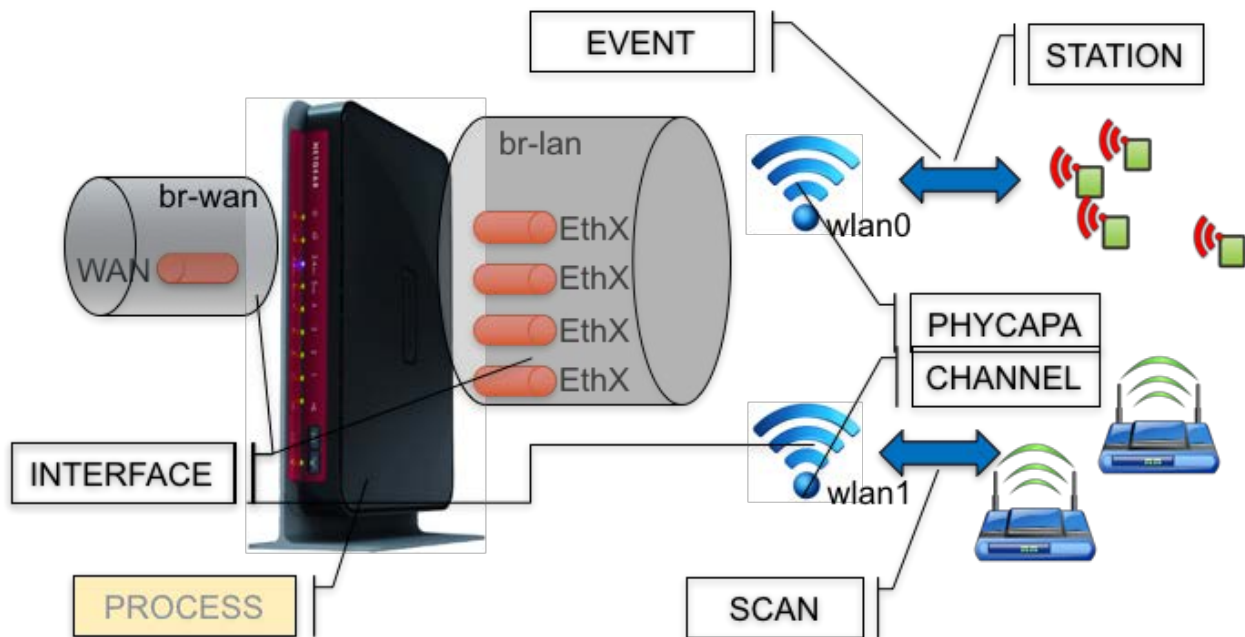


Figure 2: WCCF Agent Sensor Examples

The four primary Wi-Fi related sensors include the following functions that obtain data as described in Table 2.

Table 2: Initial Wi-Fi Sensors

Sensor	Description
Scan	Other APs and associated BSSs are identified through beacon and probe messages discovered by the AP in the wireless environment that can be received and decoded by the AP operating the WCCF agent. The data available is described in Section 5 and Appendix I.
Station	All the stations attached to each Wi-Fi radio(s) and BSS on the AP that the WCCF agent is installed on. The data available is described in Section 5 and Appendix I.
Channel	All channels are evaluated based on the AP's geographic jurisdiction settings, including the channels in primary use by the AP where the WCCF agent is installed. The data available is described in Section 5 and Appendix I.
Interface	Evaluates attributes of the WLAN interfaces. In the future, it could include information on the non-wireless WAN and LAN interfaces if required.
PhyCapa	Sensor to obtain Channel PHY layer capability information.
Event	Listener for events such as STA association request to capture the station capabilities.

Additional sensors could be developed, such as non-wireless interface statistics and device processor statistics, as shown in Figure 2. Sensors for MoCA could be developed to operate in the same architecture to identify spectrum analysis, bit-loading, and protocol statistics in the future.

4.2.1.2 WCCF Transmitter

The transmitter takes data from the processors and sends it to the receiver in the collector as shown in Figure 1. The initial protocol is REST, and the data is JSON formatted. The different JSON data structure examples are available in the software documentation [WCCFSW]. The transmitter is designed to be modular to enable different protocols to be used between the WCCF agent and collector. If required, alternative protocols in the future could include TR232, AMQP, WebPA, or one of a variety of other new bus technologies in development for IoT or 5G M2M.

4.2.1.3 WCCF Processor

The WCCF processor is a framework for software modules that can take data from single or multiple sensors to derive additional metrics or process events and workflow of the WCCF agent. For example, a calculation across Wi-Fi sensors could be completed, or deriving the channel number from the channel frequency could be done in a WCCF processor module. In the future, a processor could be used to set a policy on what data is transmitted to the collector and when.

4.2.2 WCCF Collector

The WCCF collector is the architecture component to which the data is delivered, as shown in Figure 1. Typically, this server would reside in a cloud service, or in the operator's OSS data center. This collector can be utilized as a cache that could be used to supply a "data lake," where a variety of applications and users can access the data for longer term analytics. Data can be obtained by any of these applications and stored and analyzed in those applications acting as virtual network functions (VNFs), such as a radio resource manager. The data is kept as files in an organized file system to provide flexibility on future Northbound application uses from the WCCF collector. In the future, a standardized interface can be developed to simplify data access across all technologies in the CCF [DCCFA]. The file system and structure are identified in the software documentation [WCCFSW].

4.2.2.1 WCCF Receiver

The WCCF receiver is the other end of data transfer, receiving data from a plurality of transmitters within WCCF agents in home networks. The initial protocol is REST, and the data is JSON formatted. The different data structures are defined in the software documentation [WCCFSW]. The receiver is designed to be modular to enable different protocols to be used between the WCCF agent and collector.

Data is taken from the receiver, and some formatting and storage functions make the data available to applications such as the Splunk demonstration dashboard, which is a separate module that could be replaced by or operated in parallel with a variety of applications that CableLabs may choose to develop.



Figure 4: Example Dashboard for Statistics for the AP of Interest



Figure 5: Example Dashboard of In-use and Available Channel Statistics

5 WCCF SENSOR METRICS

Table 3 identifies the key metrics as described in the CableLabs technical paper [WKPI]. Each [WKPI] metric category has a general description followed by several metrics supporting that description. Please refer to [WKPI] for more detailed discussion of each metric category. Following the [WKPI] categories, additional metrics are made available including metadata supporting the KPI categories and other related information on the wireless environment. This table also includes a set of data elements that are used to derive the KPI metrics.

For each of the data elements in Table 3, Table 4 identifies the data element and the associated base nl80211 driver API interface objects or attributes that are obtained from the OpenWrt kernel. In some instances, additional details on the derivation and calculation of the metrics are identified in Section 5.1, following Table 4. The nl80211

attributes that are currently included in the codebase, and several that could be utilized in the future, are defined in Appendix I.

Table 3: Wi-Fi KPI metrics

KPI Category	Metric	Description	Data element(s) and Calculation
PHY	General Description	Physical layer data rate used by the STA. Too low a data rate takes up too much airtime, degrading performance for other devices closer to the AP. The data rate should adapt quickly to movement towards or away from APs that influence link attenuation. The MCS index for VHT(.11ac) and HT(.11n) identify a set of modulation, MIMO, and coding rates that with the channel bandwidth can determine the throughput of the PPDU. Changes in MCS use may indicate issues with contention or interference, in addition to basic link budget coverage issues.	
PHY	staRxMcs, attacks	The modulation and coding scheme index that identifies the specific modulation and coding rate related to the bitrate for both the RX and TX channel. The MCS data is indexed differently for HT (802.11n and backward compatibility) and VHT (802.11ac). Up until VHT the MCS indexed from 0 to 31, and with VHT the index changed to 0 to 9 AND the number of spatial streams (NSS). Actual data rates also vary for a specific MCS based on the channel width and the guard interval size.	Device.WiFi.Station.HtRxMCS Device.WiFi.Station.HtTxMCS Device.WiFi.Interface.HtChannelType Device.WiFi.Station.VhtRxMCS Device.WiFi.Station.VhtTxMCS Device.WiFi.Station.VhtRxNSS Device.WiFi.Station.VhtTxNSS Device.WiFi.Interface.ChannelWidth Device.WiFi.Interface.CenterFreq1 Device.WiFi.Interface.CenterFreq2 Device.WiFi.Station.VhtTxShortGI Device.WiFi.Station.VhtRxShortGI See Section 5.1.3
PHY	staTxDataRate	The data rate reported for the station for the current physical layer based on the MCS, channel width, and Guard Interval based on the current settings. This rate can change over time as client devices move, the environment changes, or the interference changes to maintain connectivity and coverage.	Device.WiFi.Station.TxBitrate
PHY	staRxDataRate	The data rate reported for the station for the current physical layer based on the MCS, channel width, and Guard Interval based on the current settings. This rate can change over time as client devices move, the environment changes, or the interference changes to maintain connectivity and coverage.	Device.WiFi.Station.RxBitrate
RSSI	General Description	Receive Signal Strength Indicator (RSSI) is a unit-less normalized measure of RX Power level. Measure of receive power is important when low compared to that required by the MCS for a given noise level. For a given noise level, a receive power may be sufficient for TCP-based data, but insufficient for VoIP. RSS is a reasonable metric of coverage. The user may need to calibrate the value across vendors with different receiver performance for either RSSI or RSS in dBm.	

KPI Category	Metric	Description	Data element(s) and Calculation
RSSI	staRss	The Received Signal Strength (dBm) of a station measured at and reported by an AP for a specific channel. The signal can be measured for the last PDU or the average over several PDUs. Other metrics could be made available, such as RX Power in each MIMO RX chain for the last PDU, or averaged over time. Another option is to measure the average RX power for several beacon/probes from the station. Initial focus will be on the RSS of the last PDU and the average over several PDUs.	Device.WiFi.Station.Signal Device.WiFi.Station.SignalAvg See Section 5.1.3.
RSSI	scanApRss	The received signal strength of a neighboring AP (within the range of the current AP with signal strong enough to decode) based on the beacon signal for the BSS of the neighboring AP. The signal power reported is the average over several beacon frames. The number of beacon frames decoded may also be available as is the channel (number and frequency) it is operating on.	Device.WiFi.Scan.Signal See Section 5.1.1.
SNR	General Description	The ratio of Rx Power (RSS) to Noise Power for a specific station will determine which MCS can be used with low packet loss. SNR will determine capacity, and delay due to loss because of receive sensitivity issues or interference.	
SNR	staSnr	The ratio of Rx Power (RSS) to Noise Power for a specific station.	Device.WiFi.Channel.Noise, Device.WiFi.Station.Signal Device.WiFi.Station.SignalAvg See Section 5.1.1.
SNR	chanNoise	The noise level of all background noise in the channel of interest. The noise level is reported for all available channels in the current geographical jurisdiction as part of a survey of the channels from the perspective of the AP. The survey can be used to determine the best channels with the lowest noise and least amount of airtime being used. Statistics can be calculated on this metric for minimum, average, or maximum as surveys of channels are executed over time. The channel number and frequency are also helpful metadata to give context to the noise power value.	Device.WiFi.Channel.Noise See Section 5.1.1. Device.WiFi.Channel.Frequency Device.WiFi.Channel.ChannelNumber See Section 5.1.4.

KPI Category	Metric	Description	Data element(s) and Calculation
Channel Utilization and Selection	General Description	Channel Utilization can be thought of as the actual channel resource used, divided by the actual channel resource available. The resource can be based on airtime, actual traffic metrics, or spectrum in use by the existing or neighboring APs. For best use of the available spectrum, APs within range of each other should be set up on non-overlapping channels using all of the available spectrum. Co-channel use of airtime can also be managed by band and channel steering or isolation of slower rate devices to specific channels to fairly distribute the competition for channel access. Selecting no overlapping adjacent channels and reducing the level of co-channel contention can have a significant positive impact on performance. Because Automatic Channel Selection (ACS) does not work well on many APs, or only runs once at boot time, radio resource planning using an SDN controller to manage channel plans can be one of the most powerful ways to improve performance. To improve channel selection, the current state of channel use and the level of use is important. There are a variety of metrics that may be useful indicators of the level of channel utilization and for channel selection.	
Channel Utilization and Selection	chanActivityFactor	The percent of time the channel is in use for transmitting and receiving data to and from associated clients by the specific AP. Time includes data, contention, and management overhead such as beacons, probes, and other management overhead.	Device.WiFi.Channel.ChannelActiveTime, Device.WiFi.Channel.ChannelTransmitTime, Device.WiFi.Channel.ChannelReceiveTime, Device.WiFi.Channel.InUse See Section 5.1.2.
Channel Utilization and Selection	chanChannelUtil	Percentage of time the channel was occupied by the radio's own activity (activity factor) plus the activity of other radios.	Device.WiFi.Channel.ChannelActiveTime, Device.WiFi.Channel.ChannelTransmitTime, Device.WiFi.Channel.ChannelReceiveTime, Device.WiFi.Channel.ChannelBusyTime, Device.WiFi.Channel.ChannelBusyExtTime, Device.WiFi.Channel.InUse See Section 5.1.2.
Channel Utilization and Selection	chanNoAccessTime	Percentage of time that the radio was unable to transmit or receive Wi-Fi packets to/from associated clients due to Energy Detection (ED) on the channel or Clear Channel Assessment (CCA). The CCA should include both physical carrier sense and virtual carrier sense Network Allocation Vector (NAV).	Device.WiFi.Channel.ChannelActiveTime, Device.WiFi.Channel.ChannelBusyTime, Device.WiFi.Channel.ChannelBusyExtTime, Device.WiFi.Channel.InUse See Section 5.1.2.
Channel Utilization and Selection	staTxPacketRate	The rate of packets transmitted to a station.	Device.WiFi.Station.TxPackets See Section 5.1.2.
Channel Utilization and Selection	staRxPacketRate	The rate of packets received from a station.	Device.WiFi.Station.RxPackets See Section 5.1.2.
Channel Utilization and Selection	staTxDataRate	The rate of bytes transmitted to a station.	Device.WiFi.Station.TxBytes See Section 5.1.2.
Channel Utilization and Selection	staRxDataRate	The rate of bytes received from a station.	Device.WiFi.Station.RxBytes See Section 5.1.2.

KPI Category	Metric	Description	Data element(s) and Calculation
Channel Utilization and Selection	apChanCurrnetCf, apChanCurrentNum, apChanCurrentWidth	The current Center Frequency and Channel Number of the Wi-Fi channels actively being used by this AP, including the channel width and if it's a primary or secondary channel.	Device.WiFi.Channel.Frequency Device.WiFi.Channel.ChannelNumber Device.WiFi.Interface.ChannelWidth Device.WiFi.Channel.InUse
Channel Utilization and Selection	scanChanCf, scanChanNum, scanChanSelectFactor	A variety of algorithms could be used to rate the different channels for the best one to select. A variety of factors could be used to determine the desirability including: Current use by other Wi-Fi BSSs Noise levels in the channel Channel activity Bandwidth of the desired channel Linux wireless defines a simple metric based on a survey of all channels in the geographic jurisdiction that they call Interference Factor ² . This is essentially the ratio of the time the channel is busy to the time spent on the channel, multiplied by an exponential factor related to the noise level of this channel compared to the minimum noise level on all the channels.	Device.WiFi.Channel.Frequency Device.WiFi.Channel.ChannelNumber Device.WiFi.Interface.ChannelWidth Device.WiFi.Channel.InUse Device.WiFi.Channel.ChannelActiveTime, Device.WiFi.Channel.ChannelTransmitTime, Device.WiFi.Channel.ChannelReceiveTime, Device.WiFi.Channel.ChannelBusyTime, Device.WiFi.Channel.ChannelBusyExtTime, Device.WiFi.Channel.InUse
Frame Retries	General Description	If a frame (MPDU) is sent but not acknowledged, the Wi-Fi STA retries to send the frame. If a frame exceeds its maximum number of retries it fails and the frame is discarded. Since Wi-Fi is based on CSMA/CA, some retries are expected. Tracking the rate of retries and failures is useful to see if it is increasing due to traffic or interference.	
Frame Retries	staRetryRate	The rate of transmit retries to send a MPDU, based on the count over time.	Device.WiFi.Station.TxRetries
Frame Retries	staFailRate	The rate of transmit failures that discard an MPDU, based on the count over time.	Device.WiFi.Station.TxFailed
Client Capabilities	General Description	Legacy 802.11 client technologies such as 802.11bg in the 2.4 GHz band, or even 802.11a in the 5 GHz band, take up more air time than newer 802.11n in 2.4 GHz and 802.11ac in 5 GHz clients. If the goal is to optimize the performance of 802.11n and 802.11ac devices, isolating legacy devices to specific channels is one technique that may optimize performance. To use this optimization technique, the capabilities of the AP and the stations must be understood.	
Client Capabilities	staCapability	Depending on which 802.11 standards and amendments are supported, the station may be classified. If STA only supports .11a or .11b, it may be managed or isolated to a specific channel. A classification of the station's max capabilities as 802.11 legacy/a/b/g/n/ac is determined.	Device.WiFi.Event.STACapabilities.ConnectFrequency Device.WiFi.Event.STACapabilities.ConnectChannelNumber Device.WiFi.Event.STACapabilities.ConnectMode Device.WiFi.Event.STACapabilities.ConnectSignal Device.WiFi.Event.STACapabilities.Supporte dBitrate See Section 5.1.5.

² <https://wireless.wiki.kernel.org/en/users/documentation/acs>

KPI Category	Metric	Description	Data element(s) and Calculation
Client Capabilities	apChanCapability	Depending on which 802.11 standards and amendments are supported by the AP's channel, it may be classified. By removing support for certain MCS indexes, 802.11b stations that consume a lot of airtime with slow speeds are unable to connect, avoiding reduced performance for other more capable stations on the channel. This can also help improve customer QoE, avoiding very low speed connections that may be less desirable than no connectivity, or resolve "sticky client" issues when other APs with a better link budget are available.	Device.WiFi.PhyCapa.Capabilities.Frequency Device.WiFi.PhyCapa.Capabilities.ChannelNumber Device.WiFi.PhyCapa.Capabilities.MaxTxPower Device.WiFi.PhyCapa.Capabilities.ChannelWidth Device.WiFi.PhyCapa.VHTRxMCSSet.NSS.MCS, Device.WiFi.PhyCapa.VHTTxMCSSet.NSS.MCS Device.WiFi.PhyCapa.VHTShortGI Device.WiFi.PhyCapa.HTMCS See Section 5.1.5.
Other	General Description	There are a set of other metrics, not identified above, that are available and already collected by the platform and may be useful for Metadata about the BSS, general traffic metrics associated with the AP, or other Wi-Fi performance metrics. More detail on calculations will be added to this document as they are adopted in KPIs by CableLabs as needed; most are self-descriptive.	
Other	staMetadata	Client information useful for tracking client devices.	Device.WiFi.Station.APName Device.WiFi.Station.APMACAddress Device.WiFi.Station.APIndex Device.WiFi.Station.CurrentUTCTime Device.WiFi.Station.StationName Device.WiFi.Station.MACAddress Device.WiFi.Station.IPv4Address Device.WiFi.Event.EventTime Device.WiFi.Event.EventType Device.WiFi.Event.APMACAddress Device.WiFi.Event.CurrentUTCTime Device.WiFi.Event.STACapabilities.STAMAC
Other	InterfaceMetadata	WLAN interface information useful for tracking interface	Device.WiFi.Interface.APIndex Device.WiFi.Interface.APMACAddress Device.WiFi.Interface.APName Device.WiFi.Interface.CurrentUTCTime Device.WiFi.PhyCapa.APName Device.WiFi.PhyCapa.APMACAddress Device.WiFi.PhyCapa.APIndex Device.WiFi.PhyCapa.CurrentUTCTime
Other	StaMetrics	Wi-Fi data metric describing the time elapsed since last activity for this station on the AP.	Device.WiFi.Station.InactiveTime
Other	chanMetadata	Channel information useful for tracking specific channels. Other metadata like channel width and channel number are used in prior metrics.	Device.WiFi.Channel.APName Device.WiFi.Channel.APMACAddress Device.WiFi.Channel.APIndex Device.WiFi.Channel.CurrentUTCTime

KPI Category	Metric	Description	Data element(s) and Calculation
Other	scanWirelessEnvironment	The set of metrics identified by CableLabs focusing primarily on the specific channel metrics of the AP and the stations attached to those channels. Important to understanding the options for optimizing performance are details about other BSSs, including APs and stations that may be operating co-channel or in other channels closely located in the wireless environment that could be used to optimize the BSS of interest. Some of the metrics above identify the interference or impact on channel access. These scanWirelessEnvironment metrics identify the other devices that exist in the wireless environment that may be causing the interference or channel access delays. If some of these other APs and BSSs are also under the management of the operator, an overall wireless environment RRM optimization can holistically improve performance. Note: These metrics are only a subset of a comprehensive set of data available regarding the wireless environment, others could be added as needed in future releases.	Device.WiFi.Scan.APName Device.WiFi.Scan.APMACAddress Device.WiFi.Scan.APIfIndex Device.WiFi.Scan.CurrentUTCTime Device.WiFi.Scan.RemoteMACAddress Device.WiFi.Scan.Frequency Device.WiFi.Scan.SSID Device.WiFi.Scan.Signal Device.WiFi.Scan.ChannelNumber Device.WiFi.Scan.StationCnt Device.WiFi.Scan.ChannelUtil

5.1 WCCF Derived Values

Table 4: WCCF Data Elements

Data Element	Units	Description	Calculation	Ni80211 Objects
Device.WiFi.Station.HtRxMCS, Device.WiFi.Station.HtTxMCS	U8I	The MCS index as described for 802.11n including backward compatibility for older stations.	The MCS index along with the channel width and presence of a short guard interval define the rate as described in Section 5.1.3.	NL_STA_INFO_TX_BITRATE or NL_STA_INFO_RX_BITRATE: NL80211_RATE_INFO_MCS
Device.WiFi.Interface.HtChannelType	List of strings	This metric describes the possible HT channel types. These types can be used to identify the size of the channel, and where the primary and secondary channels are located.	This returns a screen descriptive of channel type, per the definitions in Appendix I.	nl80211_ATTR_WIPHY_CHANNEL_TYPE enumeration: NL80211_CHAN_NO_HT NL80211_CHAN_HT20 NL80211_CHAN_HT40MINUS NL80211_CHAN_HT40PLUS unknown
Device.WiFi.Station.VhtRxMCS, Device.WiFi.Station.VhtTxMCS	U8I	The MCS index as described for 802.11ac stations.	The MCS index along with the channel width, NSS and presence of a short guard interval define the rate as described in Section 5.1.2.	NL_STA_INFO_TX_BITRATE or NL_STA_INFO_RX_BITRATE: NL80211_RATE_INFO_VHT_MCS
Device.WiFi.Station.VhtRxNSS, Device.WiFi.Station.VhtTxNSS	U8I	The count of spatial streams for 802.11ac stations, used with MCS to estimate data rate.	The number of spatial streams is determined by the number of antennas and increases the data rate as described in Section 5.1.2.	NL_STA_INFO_TX_BITRATE or NL_STA_INFO_RX_BITRATE: NL80211_RATE_INFO_VHT_NSS

Data Element	Units	Description	Calculation	NL80211 Objects
Device.WiFi.Interface.ChannelWidth	List of strings	The width of the channel can be determined from this attribute.	A table can be used to match these attributes to the width of the channel in MHz as expected from the name. See Section 5.1.2.	NL80211_ATTR_CHANNEL_WIDTH enumeration: NL80211_CHAN_WIDTH_20_NOHT, NL80211_CHAN_WIDTH_20 NL80211_CHAN_WIDTH_40 NL80211_CHAN_WIDTH_80 NL80211_CHAN_WIDTH_80P80 NL80211_CHAN_WIDTH_160 NL80211_CHAN_WIDTH_5 NL80211_CHAN_WIDTH_10 unknown
Device.WiFi.Station.VhtTxShortGI Device.WiFi.Station.VhtRxShortGI	String	If present, indicates that the 400 nsec SGI is used.	This metric can be used to determine the data rate of the channel and the MCS as described in Section 5.1.3.	NL_STA_INFO_TX_BITRATE or NL_STA_INFO_RX_BITRATE: NL80211_RATE_INFO_SHORT_GI
Device.WiFi.Station.TxBitrate Device.WiFi.Station.RxBitrate	U32I (100 kbps)	The transmit or receive bitrate over the air, as determined by the MCS.	These metrics indicate the bitrate that should be supported for an over-the-air PDU, corrected for some physical layer overhead as described in Section 5.1.2.	NL80211_STA_INFO_TX_BITRATE enumeration: NL80211_RATE_INFO_BITRATE32 Note: If U32I is not available, there is an older U16I (100 kbps) NL80211_STA_INFO_BITRATE
Device.WiFi.Channel.Noise	(dBm)	The noise power level is measured per channel on the AP with the agent	See Section 5.1.3.	NL80211_SURVEY_INFO_NOISE
Device.WiFi.Station.Signal	U8I (dBm) for STA metrics. S32I for MBM (100*dBm). UNSPEC is an RSSI relative metric between 0 and 100	There are a variety of measures of received signal power, including the power of the last PDU or an average for a station, a specific RX receive chain, the beacon from a station, or another AP.	See Section 5.1.1 for the use of this metric in staSnr and staRSS. Other objects are defined in Appendix I for potential future use.	NL80211_STA_INFO_SIGNAL Other RSS attributes that may be of interest in the future, not expected in current release plans. NL80211_ATTR_RX_SIGNAL_DBM NL80211_STA_INFO_CHAIN_SIGNAL, NL80211_STA_INFO_CHAIN_SIGNAL_AVG, NL80211_STA_INFO_BEACON_SIGNAL_AVG, NL80211_BSS_SIGNAL_MBM, NL80211_BSS_SIGNAL_UNSPEC
Device.WiFi.Station.SignalAvg	(dBm)	The average RSS over time, not just the last PDU.	See Section 5.1.1 for the use of this metric in staSnr and staRSS.	NL80211_STA_INFO_SIGNAL_AVG
Device.WiFi.Channel.Frequency	U32I (MHz)	The center frequency of a channel. This metric can be used with Device.WiFi.Interface.ChannelWidth to describe the spectrum of the channel	The frequency is returned in MHz. To determine if the channel is currently being used by the AP, the IN_USE Object will return information. See Section 5.1.4.	nl80211_survey_info enumeration: NL80211_SURVEY_INFO_FREQUENCY, NL80211_SURVEY_INFO_IN_USE

Data Element	Units	Description	Calculation	NL80211 Objects
Device.WiFi.Channel.ChannelNumber	U8I	This is the channel number associated with the center frequency of the channel	The channel number is often how consumers and WLAN engineers think of channels, as opposed to the frequency. The channel number can be obtained from the frequency via lookup table, or can be calculated as described in Section 5.1.4.	NL80211_SURVEY_INFO_FREQUENCY
Device.WiFi.Channel.ChannelActiveTime	U32I (msec)	The amount of time the radio has spent on the channel	Obtain the data in msec directly from the NL80211 object as described in Appendix I, See Section 5.1.2.	NL80211_SURVEY_INFO_CHANNEL_TIME
Device.WiFi.Channel.ChannelTransmitTime	U32I (msec)	The amount of time the radio has spent on the channel transmitting data	Obtain the data in msec directly from the NL80211 object as described in Appendix I. See Section 5.1.2.	NL80211_SURVEY_INFO_CHANNEL_TIME_TX
Device.WiFi.Channel.ChannelReceiveTime,	U32I (msec)	The amount of time the radio has spent on the channel receiving data.	Obtain the data in msec directly from the NL80211 object as described in Appendix I. See Section 5.1.2.	NL80211_SURVEY_INFO_CHANNEL_TIME_RX
Device.WiFi.Channel.ChannelBusyTime,	U32I (msec)	The amount of time the radio has spent on the channel, but noticed the channel was busy and could not initiate communication if it wanted to. I believe this includes both physical CCA based on ED and also CS based on NAV.	Obtain the data in msec directly from the NL80211 object as described in Appendix I. Based on testing to be done on 40 MHz wide .11n channels, the EXT_BUSY may need to be considered. See Section 5.1.2.	NL80211_SURVEY_INFO_CHANNEL_TIME_BUSY NL80211_SURVEY_INFO_CHANNEL_TIME_EXT_BUSY
Device.WiFi.Channel.In Use	String	If this channel is currently in use by the radio, the presence of this object confirms it.	Detect its presence	NL80211_SURVEY_INFO_IN_USE
Device.WiFi.Station.Tx Packets	U32I	The number of transmitted MPDUs to this station	Leverage the count as described for the NL80211 object described in Appendix I	nl80211_sta_info enumeration: NL80211_STA_INFO_TX_PACKETS
Device.WiFi.Station.Rx Packets	U32I	The number of MPDUs received from this station	Leverage the count as described for the NL80211 object described in Appendix I	nl80211_sta_info enumeration: NL80211_STA_INFO_RX_PACKETS
Device.WiFi.Station.Tx Bytes	U64I	The number of bytes transmitted to this station	Leverage the count as described for the NL80211 object described in Appendix I	nl80211_sta_info enumeration: NL80211_STA_INFO_TX_BYTES NL80211_STA_INFO_TX_BYTES64
Device.WiFi.Station.Rx Bytes	U64I	The number of bytes received from this station	Leverage the count as described for the NL80211 object described in Appendix I	nl80211_sta_info enumeration: NL80211_STA_INFO_RX_BYTES NL80211_STA_INFO_RX_BYTES64

Data Element	Units	Description	Calculation	NL80211 Objects
Device.WiFi.Station.Tx Retries	U32I	Total retries (not including first attempt) to transmit MPDUs to this station	Leverage the count as described for the NL80211 object described in Appendix I. Device.WiFi.Station.TxRetries = NL80211_STA_INFO_TX_RETRIES	nl80211_sta_info enumeration: NL80211_STA_INFO_TX_RETRIES
Device.WiFi.Station.Tx Failed	U32I	The number of MPDUs that failed in transmission to the station after retries	Leverage the count as described for the NL80211 object described in Appendix I. Device.WiFi.Station.TxFailed = NL80211_STA_INFO_TX_FAILED	nl80211_sta_info enumeration: NL80211_STA_INFO_TX_FAILED
Device.WiFi.Station.AP Name	String	The name of the interface the station is connected to on the AP.	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_IFNAME
Device.WiFi.Station.AP MACAddress	6 octets	The MAC address of the STA	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_MAC
Device.WiFi.Station.AP IfIndex	U32I	Interface Index of the WLAN interface where the station is attached	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_IFINDEX
Device.WiFi.Station.CurrentUTCTime	UTC time	Time data was obtained for the specific station	Convert UTC into local time as needed. For example, to convert to MST, subtract 7 hours. To convert into MDT, subtract 6 hours.	From software
Device.WiFi.Station.StationName	String	A name that can be obtained through name resolution	A configuration of the WCCF or based on DNS lookup	Comes from other method
Device.WiFi.Station.MACAddress	6 octets	The MAC address of the station attached to this AP	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_MAC
Device.WiFi.Station.IPv4Address	IP Address	IP Address of the station, obtained through DHCP		Comes from other method
Device.WiFi.Station.InactiveTime	U32I (msec)	The time since the last activity on the channel.	As described for the NL80211 object described in Appendix I.	nl80211_sta_info enumeration: NL80211_STA_INFO_INACTIVE_TIME
Device.WiFi.Channel.AP MACAddress	6 octets	The MAC address or BSSID of the channel on the AP.	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_MAC
Device.WiFi.Channel.AP IfIndex	Integer	The ifIndex for the channel in use on the AP.	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_IFINDEX
Device.WiFi.Channel.CurrentUTCTime	UTC time	Time data was obtained for the specific Channel	Convert UTC into local time as needed. For example, to convert to MST subtract 7 hours. To convert into MDT, subtract 6 hours.	From software
Device.WiFi.Scan.AP Name	String	The ifName of the AP radio that is performing the scan of the wireless environment.	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_IFNAME

Data Element	Units	Description	Calculation	NL80211 Objects
Device.WiFi.Scan.APM ACAddress	6 octets	The MAC address of the AP's BSSID that executed the scan of the wireless environment.	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_MAC
Device.WiFi.Scan.APIfl ndex	Integer	The ifIndex of the AP's WLAN interface that executed the scan of the wireless environment.	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_IFINDEX
Device.WiFi.Scan.Curr entUTCtime	UTC time	Time data was obtained for the specific scan of the wireless environment.	Convert UTC into local time as needed. For example, to convert to MST subtract 7 hours. To convert into MDT subtract 6 hours.	From software
Device.WiFi.Scan.Rem oteMACAddress	6 octets	The MAC address (BSSID) of another remote BSS / AP that is close enough to this AP to decode the beacon.	As described for the NL80211 object described in Appendix I.	NL80211_BSS_BSSID
Device.WiFi.Scan.Freq uency	U32I (MHz)	The frequency channel where the AP scan decoded the beacon of the remote AP.	As described for the NL80211 object described in Appendix I. See Section 5.1.4 for details on getting channel from frequency.	NL80211_BSS_FREQUENCY
Device.WiFi.Scan.SSI D	String	The SSID used for the remote AP discovered in the scan of the wireless environment.	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_SSID
Device.WiFi.Scan.Sign al	dBm	The received signal strength of the beacon from the remote AP discovered in the scan	NL80211_BSS_SIGNAL_MBM / 100 (get to dBm from MBM)	NL80211_BSS_SIGNAL_MBM
Device.WiFi.Scan.Cha nnelNumber	U8I	This is the channel number associated with the center frequency of the channel where the remote AP was discovered in the wireless environment	The channel number is often how consumers and WLAN engineers think of channels as opposed to the frequency. The channel number can be obtained from the frequency via lookup table, or can be calculated as described in Section 5.1.4.	NL80211_BSS_FREQUENCY
Device.WiFi.Scan.Stati onCnt	Integer	The count of stations attached to the BSS of the remote AP	If available, the station count from the remote AP	?
Device.WiFi.Scan.Cha nnelUtil	%	Channel Utilization	?	?
Device.WiFi.Interface. CurrentUTCtime	UTC time	Time data was obtained for the specific scan of the wireless environment.	Convert UTC into local time as needed. For example, to convert to MST subtract 7 hours. To convert into MDT, subtract 6 hours.	From software
Device.WiFi.Interface. CenterFreq1	MHz	Frequency if 2.11n channels	Used with Interface Channel width if a .11n channel with primary and secondary	NL80211_ATTR_CENTER_FREQ1
Device.WiFi.Interface. CenterFreq2	MHz	Frequency if 2 .11n channels	Used with Interface Channel width if a .11n channel with primary and secondary	NL80211_ATTR_CENTER_FREQ2

Data Element	Units	Description	Calculation	NL80211 Objects
Device.WiFi.Interface.APIndex	Integer	The ifindex of the WLAN interface	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_IFINDEX
Device.WiFi.Interface.APMACAddress	6 octets	The MAC address of the WLAN interface	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_MAC
Device.WiFi.Interface.APName	String	The APNAME for the WLAN interface	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_IFINDEX
Device.WiFi.Channel.APMACAddress	6 octets	MAC address of the AP Channel	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_MAC
Device.WiFi.Channel.APName	String	AP NAME for the Channel	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_IFINDEX
Device.WiFi.Event.STACapabilities.ConnectChannelNumber	Integer	This is the channel number the association request is transmitted on	The channel number is often how consumers and WLAN engineers think of channels, as opposed to the frequency. The channel number can be obtained from the frequency via lookup table, or can be calculated as described in Section 5.1.4.	From Association Request Parsing
Device.WiFi.Event.STACapabilities.ConnectFrequency	MHz	This is the frequency of the channel the association request is transmitted on	Number is in MHz	From Association Request Parsing
Device.WiFi.Event.STACapabilities.ConnectMode	String	This is the mode of the association request	String describing mode of connection such as 11a or 11b	From Association Request Parsing
Device.WiFi.Event.STACapabilities.ConnectSignal	dBm	This is the signal power of the association request	It's in dBm	From Association Request Parsing
Device.WiFi.Event.STACapabilities.SupportedBitrate	Array of float.1	This is an array listing all the bitrates supported by the station as advertised in the association request	Units of Mbps, See Section 5.1.3	From Association Request Parsing
Device.WiFi.Event.STACapabilities.STAMAC	6 octets	This is the STAs MAC Address that is making the association request	NA: separated bytes	From Association Request Parsing
Device.WiFi.Event.APMACAddress	6 octets	This is the AP WLAN MAC address where the association request was received	NA: separated bytes	From Association Request Parsing
Device.WiFi.Event.CurrentUTCTime	UTC time	This is the UTC time of the event data set creation	Convert UTC into local time as needed. For example, to convert to MST subtract 7 hours. To convert into MDT, subtract 6 hours.	From software
Device.WiFi.Event.EventTime	UTC time	This is the UTC time the event was detected	Convert UTC into local time as needed. For example, to convert to MST subtract 7 hours. To convert into MDT, subtract 6 hours.	From software
Device.WiFi.Event.EventType	String	This is the type of event, for R3, the only type is association request	Simple string such as association request, in future may include additional events	From software based on sensor configuration. Association Request is current type

Data Element	Units	Description	Calculation	NL80211 Objects
Device.WiFi.PhyCapa. APIfIndex	Integer	This is the ifIndex of the WLAN interface the Channel capabilities belong to	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_IFINDEX
Device.WiFi.PhyCapa. APMACAddress	6 octets	This is the MAC address of the WLAN interface these capabilities are associated with	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_MAC
Device.WiFi.PhyCapa. APName	String	This is the AP name of the WLAN interface the capabilities are associated with	As described for the NL80211 object described in Appendix I.	NL80211_ATTR_IFINDEX
Device.WiFi.PhyCapa. Capabilities.ChannelNumber	Array of Integer	These are the channel numbers that the WLAN is capable of using	The channel number is often how consumers and WLAN engineers think of channels, as opposed to the frequency. The channel number can be obtained from the frequency via lookup table, or can be calculated as described in Section 5.1.3 and 5.1.4, and see Section 5.1.5	NL80211_FREQUENCY_ATTR_FREQ
Device.WiFi.PhyCapa. Capabilities.ChannelWidth	Array of MHz	These are the channel bandwidths that the WLAN is capable of using	Array of strings describing channel width. Could be easily converted to integer. See Section 5.1.3 and see Section 5.1.5	NL80211_FREQUENCY_ATTR_NO_20MHZ, NL80211_FREQUENCY_ATTR_NO_HT40_MINUS, NL80211_FREQUENCY_ATTR_NO_HT40_PLUS, NL80211_FREQUENCY_ATTR_NO_80MHZ, NL80211_FREQUENCY_ATTR_NO_160MHZ
Device.WiFi.PhyCapa. Capabilities.MaxTxPower	dBm	This is the maximum Tx Power this WLAN interface is capable of.	NA	NL80211_FREQUENCY_ATTR_MAX_TX_POWER
Device.WiFi.PhyCapa. CurrentUTCtime	UTC time	This is the time the channel capabilities were collected	Convert UTC into local time as needed. For example, to convert to MST subtract 7 hours. To convert into MDT, subtract 6 hours.	From software
Device.WiFi.PhyCapa. HTMCS	Array of Integer	This is the list of MCS indexes supported for 802.11n functionality for this WLAN interface	Array of integers indicating MCS. See Section 5.1.3 and see Section 5.1.5	NL80211_BAND_ATTR_HT_MCS_SET
Device.WiFi.PhyCapa. VHTRxMCSSet.NSS.MCS, Device.WiFi.PhyCapa. VHTTxMCSSet.NSS.MCS	2D Array of Integer	This is a two-dimensional array of the 802.11ac MCS indexes supported for each supported number of spatial streams for this WLAN interface	Array of integers indicating MCS. See Section 5.1.3 and see Section 5.1.5	NL80211_BAND_ATTR_VHT_CPA, NL80211_BAND_ATTR_VHT_MCS_SET
Device.WiFi.PhyCapa. VHTShortGI	String	This indicates whether the WLAN interface is capable of using the short guard interval	See Section 5.1.3	NL80211_BAND_ATTR_VHT_CPA
Device.WiFi.PhyCapa. Capabilities.Frequency	MHz	These are the channel frequencies that the WLAN is capable of using.	See Section 5.1.3	NL80211_FREQUENCY_ATTR_FREQ, NL80211_FREQUENCY_ATTR_MAX, NL80211_FREQUENCY_ATTR_DISABLED

5.1.1 STA Signal to Noise Ratio (SNR) and RSS

SNR is the metric that describes the wireless channel quality and directly impacts the capacity of the link. A table of approximate signal to noise ratios required to support MCS indexes is shown in Figure 7. The higher the SNR, the higher the MCS and capacity of the wireless link.

Signal to noise ratio requires a measure from the channel survey to find the noise level in the channel, and it also requires the signal level from the STA. There are a variety of different Received Signal Strength Metrics as described in Table 3 and Table 4. These include the RX Power for the station's traffic, the power for each spatial stream receive chain, and the power in the beacons. Power is available for the last PDU and for an average.

The average is based on an exponential weighted moving average (EWMA) with a factor of 1024 and weight of 8, based on a patch in August 2015 to mac80211.

For the signal in the signal to noise ratio metric, we will use the attribute:

```
Device.WiFi.Station.Signal = NL80211_STA_INFO_SIGNAL /* units of dBm */
```

For the noise, we will use the noise power estimate from the channel survey:

```
Device.WiFi.Channel.Noise = NL80211_SURVEY_INFO_NOISE /*units of dBm */
```

Signal-to-Noise Ratio is calculated using these attributes:

```
staSnr = Device.WiFi.Station.SignalAvg - Device.WiFi.Channel.Noise /* units  
of dB*/
```

```
staRss = Device.WiFi.Station.SignalAvg
```

```
scanApRss = Device.WiFi.Scan.Signal /* This is data element is defined in  
Table 4*/
```

5.1.2 Channel Utilization, Channel Selection, and WLAN Traffic Metrics

Several metrics are defined as a rate; for example, a data rate in bits per second. When describing a time-based rate, the term **DeltaT** means the number of seconds for which the data was collected.

Several of these items are based on the Channel Survey, where the AP looks at all the available channels and returns statistics for the results of each channel. The currently used channel is identified by

```
Device.WiFi.Channel.InUse = NL80211_SURVEY_INFO_IN_USE
```

The currently used channel can be useful to evaluate the specific channels used by the AP of interest, so is a relevant way to classify all the channel metrics.

The `Device.WiFi.Channel.ChannelActiveTime` as defined here is the total time the radio resides on the channel of interest.

```
chanActivityFactor = (Device.WiFi.Channel.ChannelTransmitTime +  
Device.WiFi.Channel.ChannelReceiveTime) /  
Device.WiFi.Channel.ChannelActiveTime
```

Where

```
Device.WiFi.Channel.ChannelActiveTime = NL80211_SURVEY_INFO_CHANNEL_TIME
Device.WiFi.Channel.ChannelTransmitTime = NL80211_SURVEY_INFO_CHANNEL_TIME_TX
Device.WiFi.Channel.ChannelReceiveTime = NL80211_SURVEY_INFO_CHANNEL_TIME_RX
```

```
chanChannelUtil = (Device.WiFi.Channel.ChannelTransmitTime +
Device.WiFi.Channel.ChannelReceiveTime + Device.WiFi.Channel.ChannelBusyTime)
/ Device.WiFi.Channel.ChannelActiveTime
```

Where

```
Device.WiFi.Channel.ChannelBusyTime = NL80211_SURVEY_INFO_CHANNEL_TIME_BUSY
```

Note we will need to do some testing with 40 MHz 802.11n channels to see if we also need to aggregate time on the extension channel (NL80211_SURVEY_INFO_CHANNEL_TIME_EXT_BUSY) to get the correct calculation for the full channel.

```
chanNoAccessTime = Device.WiFi.Channel.ChannelBusyTime /
Device.WiFi.Channel.ChannelActiveTime
```

```
staTxPacketRate = Device.WiFi.Station.TxPackets / DeltaT
staRxPacketRate = Device.WiFi.Station.RxPackets / DeltaT
staTxDataRate = Device.WiFi.Station.TxBytes / DeltaT
staRxDataRate = Device.WiFi.Station.RxBytes / DeltaT
```

Where

```
Device.WiFi.Station.TxPackets = NL80211_STA_INFO_TX_PACKETS
Device.WiFi.Station.RxPackets = NL80211_STA_INFO_RX_PACKETS
Device.WiFi.Station.TxBytes = NL80211_STA_INFO_TX_BYTES64 /* if not available
try NL80211_STA_INFO_TX_BYTES */
Device.WiFi.Station.RxBytes = NL80211_STA_INFO_RX_BYTES64 /* if not available
try NL80211_STA_INFO_RX_BYTES */
```

For channel selection, it's important to know what channels are in use, and the usage of the other available channels. The following metrics were defined in Table 3 and Table 4.

These are for the current channel in use by the AP of interest as described above.

```

apChanCurrnetCf = Device.WiFi.Channel.Frequency

apChanCurrentNum = See Section 5.1.4

apChanCurrentWidth = Device.WiFi.Interface.ChannelWidth

```

where

```

Device.WiFi.Channel.Frequency = NL80211_SURVEY_INFO_FREQUENCY
Device.WiFi.Interface.ChannelWidth = X /*MHz, X is a value associated with
one of an enumeration of nl80211_chan_width, which is described in Appendix I
and will include one of the enumerations in 5 */

```

Table 5: nl80211_chan_width Enumeration Mapping

Enumeration	X = Channel Width (MHz)
NL80211_CHAN_WIDTH_20_NOHT,	20 /*Pre-802.11n 20 MHz channel */
NL80211_CHAN_WIDTH_20,	20
NL80211_CHAN_WIDTH_40,	40
NL80211_CHAN_WIDTH_80,	80
NL80211_CHAN_WIDTH_80P80,	160 /*80 + 80 = 160 deprecated for some HW platforms only reports 160*/
NL80211_CHAN_WIDTH_160,	160
NL80211_CHAN_WIDTH_5,	5 /* for older Legacy 802.11 */
NL80211_CHAN_WIDTH_10	10 /* for older Legacy 802.11 */

The ChanSelectFactor is intended to be a metric that describes the desirability of all the channels that the AP of interest could use as an alternative to its current channels, and is a function of the activity in the channel and its overall use and noise level.

```

scanChanCf = Device.WiFi.Channel.Frequency

scanChanNum = See Section 5.1.4

scanChanSelectFactor = /* To be defined after discussion with CableLabs,
or suggest Interference Factor from Linux wireless as a first pass. The
project focused on "Sticky AP" has some input or earlier RRM work at CL may
have some input */

```

5.1.3 MCS Index for HT and VHT

Wireless Engineers often think of channel PHY capacity, coverage, and implications for airtime in terms of Modulation and Coding Scheme (MCS) index. 802.11n and earlier clients via backward compatibility have MCS indexes that increase with the number of spatial streams from 0 up to 31. There are actually up to 78 indicies defined in 802.11-2012, but many are optional and infrequently implemented.³ With 802.11ac, the MCS index always varies between 0 and 9, and the data rate is adjusted based on the number of spatial streams. For both MCS index approaches, the data rate varies as expected, based on channel width and the size of the guard interval (GI). The 802.11n specification introduced a short guard interval (SGI) of 400 nsec vs. 800 nsec for the regular guard interval. The guard interval is a cyclic prefix type gap between OFDM symbols. Figure 6 shows a table of MCS index and

data rates.³ These data rates, while reduced for some PHY overhead, are higher than the actual information data rate over time due to other time-based and management overhead factors.

The MCS can be adjusted by the system based on a number of transmit retries, assuming there are signal sensitivity coverage issues or interference issues that can be resolved with a more robust modulation and coding configuration. One estimate of SNR required for the different MCS indexes is shown in Figure 7; note that this SNR to MCS mapping depends a lot on the quality of the radio chain such as Noise Figure (NF) and amplifier linearity and type of noise or interference. As a result, the SNR to MCS mapping may vary for different devices. Devices could do better or worse than these numbers based on implementation.

As can be seen in Figure 6, several attributes are useful for understanding the MCS index including channel width, number of spatial streams, and guard interval. These data are made available in WCCF based on the units described in Table 4 and interpreted per Figure 6 and Figure 7.

The data elements are calculated from NL80211 objects as described below.

```
Device.WiFi.Station.HtRxMCS = NL80211_RATE_INFO_MCS /*see column 1 Figure 6*/
Device.WiFi.Station.HtTxMCS = NL80211_RATE_INFO_MCS /*see column 1 Figure 6*/
Device.WiFi.Station.VhtRxMCS = NL80211_RATE_INFO_VHT_MCS /*see column 2
Figure 6*/
Device.WiFi.Station.VhtTxMCS = NL80211_RATE_INFO_VHT_MCS /*see column 2
Figure 6*/
Device.WiFi.Station.VhtRxNSS = NL80211_RATE_INFO_VHT_NSS /see column 3 Figure
6*/
Device.WiFi.Station.VhtTxNSS = NL80211_RATE_INFO_VHT_NSS /see column 3 Figure
6*/
Device.WiFi.Interface.ChannelWidth = See definition in Section 5.1.2/*see
columns in Figure 6 for each bandwidth*/
Device.WiFi.Station.VhtTxShortGI = NL80211_RATE_INFO_SHORT_GI
Device.WiFi.Station.VhtRxShortGI = NL80211_RATE_INFO_SHORT_GI /see bandwidth
columns in Figure 6 that are split into No SGI and SGI, this field indicates
the SGI column is applicable for the channel*/
Device.WiFi.Station.TxBitrate = NL80211_RATE_INFO_BITRATE32 /* if not
available try NL80211_STA_INFO_TX_BITRATE, 32 bit is preferred */
Device.WiFi.Station.RxBitrate = NL80211_RATE_INFO_BITRATE32 /*These bit rates
should correlate with those identified in Figure 6.
```

³ <http://www.wlanpros.com/mcs-index-802-11n-802-11ac-chart-3/>, <http://mcsindex.com/>, for more specific detail see [802.11-2012](#) section 20.3.5, and tables 20-30 through 20-44

Then the Metrics are obtained from the Data elements as follows.

```

if Device.WiFi.Station.VhtRxMCS is present then
    staRxMcs = WiFi.Station.VhtRxMCS /* note you will also need
Device.WiFi.Station.VhtRxNSS, Device.WiFi.Interface.ChannelWidth and
Device.WiFi.Station.VhtRxShortGI to determine speed */
else
    staRxMcs = Device.WiFi.Station.HtRxMCS /*See speed in table */

if Device.WiFi.Station.VhtTxMCS is present then
    staTxMcs = WiFi.Station.VhtTxMCS /* note you will also need
Device.WiFi.Station.VhtTxNSS, Device.WiFi.Interface.ChannelWidth and
De Device.WiFi.Station.VhtTxShortGI to determine speed */
else
    staTxMcs = Device.WiFi.Station.HtTxMCS /*See speed in table */

staTxDataRate = Device.WiFi.Station.TxBitrate
staRxDataRate = Device.WiFi.Station.RxBitrate

```

MCS Index - 802.11n and 802.11ac

					802.11n		802.11ac					
HT MCS Index	VHT MCS Index	Spatial Streams	Modulation	Coding	20MHz		40MHz		80MHz		160MHz	
					Data Rate No SGI	Data Rate SGI	Data Rate No SGI	Data Rate SGI	Data Rate No SGI	Data Rate SGI	Data Rate No SGI	Data Rate SGI
0	0	1	BPSK	1/2	6.5	7.2	13.5	15	29.3	32.5	58.5	65
1	1	1	QPSK	1/2	13	14.4	27	30	58.5	65	117	130
2	2	1	QPSK	3/4	19.5	21.7	40.5	45	87.8	97.5	175.5	195
3	3	1	16-QAM	1/2	26	28.9	54	60	117	130	234	260
4	4	1	16-QAM	3/4	39	43.3	81	90	175.5	195	351	390
5	5	1	64-QAM	2/3	52	57.8	108	120	234	260	468	520
6	6	1	64-QAM	3/4	58.5	65	121.5	135	263.3	292.5	526.5	585
7	7	1	64-QAM	5/6	65	72.2	135	150	292.5	325	585	650
	8	1	256-QAM	3/4	78	86.7	162	180	351	390	702	780
	9	1	256-QAM	5/6	n/a	n/a	180	200	390	433.3	780	866.7
8	0	2	BPSK	1/2	13	14.4	27	30	58.5	65	117	130
9	1	2	QPSK	1/2	26	28.9	54	60	117	130	234	260
10	2	2	QPSK	3/4	39	43.3	81	90	175.5	195	351	390
11	3	2	16-QAM	1/2	52	57.8	108	120	234	260	468	520
12	4	2	16-QAM	3/4	78	86.7	162	180	351	390	702	780
13	5	2	64-QAM	2/3	104	115.6	216	240	468	520	936	1040
14	6	2	64-QAM	3/4	117	130.3	243	270	526.5	585	1053	1170
15	7	2	64-QAM	5/6	130	144.4	270	300	585	650	1170	1300
	8	2	256-QAM	3/4	156	173.3	324	360	702	780	1404	1560
	9	2	256-QAM	5/6	n/a	n/a	360	400	780	866.7	1560	1733.3
16	0	3	BPSK	1/2	19.5	21.7	40.5	45	87.8	97.5	175.5	195
17	1	3	QPSK	1/2	39	43.3	81	90	175.5	195	351	390
18	2	3	QPSK	3/4	58.5	65	121.5	135	263.3	292.5	526.5	585
19	3	3	16-QAM	1/2	78	86.7	162	180	351	390	702	780
20	4	3	16-QAM	3/4	117	130	243	270	526.5	585	1053	1170
21	5	3	64-QAM	2/3	156	173.3	324	360	702	780	1404	1560
22	6	3	64-QAM	3/4	175.5	195	364.5	405	n/a	n/a	1579.5	1755
23	7	3	64-QAM	5/6	195	216.7	405	450	877.5	975	1755	1950
	8	3	256-QAM	3/4	234	260	486	540	1053	1170	2106	2340
	9	3	256-QAM	5/6	260	288.9	540	600	1170	1300	n/a	n/a

Figure 6: MCS Index and Data Rates

MCS Value Achieved by Clients at Various Signal to Noise Ratio Levels (SNR)

Protocol	Channel	1	2	3	4	5	6	7	8	9	10	Modulation Key
802.11b	20MHz	None	None	None	MCS 0	MCS 0	MCS 0	MCS 1	MCS 1	MCS 1	MCS 1	
802.11a/g	20MHz	None	MCS 0	MCS 0	MCS 1	MCS 2	MCS 2	MCS 2	MCS 2	MCS 3	MCS 3	None = Grey
802.11n	20MHz	None	MCS 0	MCS 0	MCS 0	MCS 1	MCS 1	MCS 1	MCS 1	MCS 2	MCS 2	BPSK = Red
802.11n	40MHz	None	None	None	None	MCS 0	MCS 0	MCS 0	MCS 1	MCS 1	MCS 1	QPSK = Orange
802.11ac	20MHz	None	MCS 0	MCS 0	MCS 0	MCS 1	MCS 1	MCS 1	MCS 1	MCS 2	MCS 2	16-QAM = Yellow
802.11ac	40MHz	None	None	None	None	MCS 0	MCS 0	MCS 0	MCS 1	MCS 1	MCS 1	64-QAM = Blue
802.11ac	80MHz	None	None	None	None	None	None	None	MCS 0	MCS 0	MCS 0	256-QAM = Green
802.11ac	160MHz	None	None	None	None	None	None	None	None	None	None	
SNR in dB		11	12	13	14	15	16	17	18	19	20	802.11 Type Key
802.11b	20MHz	MCS 2	MCS 2	MCS 2	MCS 2	MCS 2	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	
802.11a/g	20MHz	MCS 4	MCS 4	MCS 4	MCS 4	MCS 5	MCS 5	MCS 5	MCS 6	MCS 6	MCS 7	802.11b
802.11n	20MHz	MCS 3	MCS 3	MCS 3	MCS 3	MCS 4	MCS 4	MCS 4	MCS 5	MCS 5	MCS 6	802.11a/g
802.11n	40MHz	MCS 1	MCS 2	MCS 2	MCS 3	MCS 3	MCS 3	MCS 3	MCS 4	MCS 4	MCS 4	802.11n
802.11ac	20MHz	MCS 3	MCS 3	MCS 3	MCS 3	MCS 4	MCS 4	MCS 4	MCS 5	MCS 5	MCS 6	802.11ac
802.11ac	40MHz	MCS 1	MCS 2	MCS 2	MCS 3	MCS 3	MCS 3	MCS 3	MCS 4	MCS 4	MCS 4	
802.11ac	80MHz	MCS 1	MCS 1	MCS 1	MCS 1	MCS 2	MCS 2	MCS 3	MCS 3	MCS 3	MCS 3	
802.11ac	160MHz	MCS 0	MCS 0	MCS 0	MCS 1	MCS 1	MCS 1	MCS 1	MCS 2	MCS 2	MCS 3	
SNR in dB		21	22	23	24	25	26	27	28	29	30	
802.11b	20MHz	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	
802.11a/g	20MHz	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	
802.11n	20MHz	MCS 6	MCS 6	MCS 6	MCS 6	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	
802.11n	40MHz	MCS 5	MCS 5	MCS 6	MCS 6	MCS 6	MCS 6	MCS 6	MCS 7	MCS 7	MCS 7	
802.11ac	20MHz	MCS 6	MCS 6	MCS 6	MCS 6	MCS 7	MCS 7	MCS 7	MCS 7	MCS 8	MCS 8	
802.11ac	40MHz	MCS 5	MCS 5	MCS 6	MCS 6	MCS 6	MCS 6	MCS 6	MCS 7	MCS 7	MCS 7	
802.11ac	80MHz	MCS 4	MCS 4	MCS 4	MCS 5	MCS 5	MCS 6	MCS 6	MCS 6	MCS 6	MCS 6	
802.11ac	160MHz	MCS 3	MCS 3	MCS 3	MCS 4	MCS 4	MCS 4	MCS 5	MCS 5	MCS 6	MCS 6	
SNR in dB		31	32	33	34	35	36	37	38	39	40	
802.11b	20MHz	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	
802.11a/g	20MHz	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	
802.11n	20MHz	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	
802.11n	40MHz	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	
802.11ac	20MHz	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	
802.11ac	40MHz	MCS 7	MCS 8	MCS 8	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	
802.11ac	80MHz	MCS 7	MCS 7	MCS 7	MCS 7	MCS 8	MCS 8	MCS 9	MCS 9	MCS 9	MCS 9	
802.11ac	160MHz	MCS 6	MCS 6	MCS 6	MCS 7	MCS 7	MCS 7	MCS 7	MCS 8	MCS 8	MCS 9	
SNR in dB		41	42	43	44	45	46	47	48	49	50	
802.11b	20MHz	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	MCS 3	
802.11a/g	20MHz	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	
802.11n	20MHz	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	
802.11n	40MHz	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	MCS 7	
802.11ac	20MHz	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	
802.11ac	40MHz	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	
802.11ac	80MHz	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	
802.11ac	160MHz	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	MCS 9	

Figure 7: MCS Value Achieved by Clients at Various Signal to Noise Ratio (SNR)

5.1.4 Frequency to Channel Conversion

The actual channel number is calculated in the current Linux wireless user space applications. For example in the iw 4.0, there are functions in util.c to convert frequencies to channel numbers and channel numbers to frequencies. This process is similar for WCCF. Another option is to do a lookup based on Table 6 and Table 7. Note: Not all channels are available in all regulatory jurisdictions around the world.

Device.WiFi.Channel.Frequency = See Section 5.1.2
 Device.WiFi.Channel.ChannelNumber = Based on Device.WiFi.Channel.Frequency
 look up in Table 6 or Table 7, or calculate per example in Section 5.1.4.1.

Table 6: 2.4 GHz Center Frequency (MHz) to Channel Lookup

Device.WiFi.Channel.ChannelNumber, Device.WiFi.Scan.ChannelNumber	Device.WiFi.Channel.Frequency, Device.WiFi.Scan.Frequency
1	2412
2	2417
3	2422
4	2427
5	2432
6	2437
7	2442
8	2447
9	2452
10	2457
11	2462
12	2467
13	2472
14	2484

Table 7: 5 GHz Center Frequency (MHz) to Channel Lookup

Device.WiFi.Channel.ChannelNumber, Device.WiFi.Scan.ChannelNumber	Device.WiFi.Channel.Frequency, Device.WiFi.Scan.Frequency
36	5180
40	5200
44	5220
48	5240
52	5260
56	5280
60	5300
64	5320
100	5500
104	5520
108	5540
112	5560
116	5580
120	5600
124	5620
128	5640
132	5660
136	5680
140	5700
149	5745
153	5765

Device.WiFi.Channel.ChannelNumber, Device.WiFi.Scan.ChannelNumber	Device.WiFi.Channel.Frequency, Device.WiFi.Scan.Frequency
157	5785
161	5805
165	5825

5.1.4.1 Code Examples Converting Frequency to Channel Number

```

int ieee80211_channel_to_frequency(int chan, enum nl80211_band band)
{
    /see 802.11 17.3.8.3.2 and Annex J
    there are overlapping channel numbers in 5 GHz and 2 GHz bands */
    if (chan <= 0)
        return 0; /not supported */
    switch (band) {
    case NL80211_BAND_2_GHZ:
        if (chan == 14)
            return 2484;
        else if (chan < 14)
            return 2407 + chan * 5;
        break;
    case NL80211_BAND_5_GHZ:
        if (chan >= 182 && chan <= 196)
            return 4000 + chan * 5;
        else
            return 5000 + chan * 5;
        break;
    case NL80211_BAND_60_GHZ:
        if (chan < 5)
            return 56160 + chan * 2160;
        break;
    default:
        ;
    }
    return 0; /not supported */
}

int ieee80211_channel_to_frequency(int chan, enum nl80211_band band)
{
    /see 802.11 17.3.8.3.2 and Annex J
    there are overlapping channel numbers in 5 GHz and 2 GHz bands */
    if (chan <= 0)
        return 0; /not supported */
    switch (band) {
    case NL80211_BAND_2_GHZ:
        if (chan == 14)
            return 2484;
        else if (chan < 14)

```

```

        return 2407 + chan 5;
    break;
case NL80211_BAND_5_GHZ:
    if (chan >= 182 && chan <= 196)
        return 4000 + chan 5;
    else
        return 5000 + chan 5;
    break;
case NL80211_BAND_60_GHZ:
    if (chan < 5)
        return 56160 + chan 2160;
    break;
default:
    ;
}
return 0; /*not supported */
}

```

5.1.5 Station and Channel Capabilities

To determine if a station should be managed in a special way because it uses older technology such as .11b in the 2.4 GHz band, or .11a in the 5 GHz band, then evaluate as described in this section. There may be other ways to get this data, and this section may be updated as the code is developed. The station capability may impact performance of the network. For example, if the station is far away from the AP and it can only operate at the lowest MCS bitrates, the amount of time it takes to send a packet is significantly increased, reducing the airtime available for other stations.

For effective Radio Resource Management (RRM), it is essential to know the capabilities of the WLAN channels on each AP, including which channels and bands are supported at what power levels, and channel bandwidth. Other attributes such as number of spatial streams can be used to understand the APs full capacity capabilities.

5.1.5.1 Station

Using the Supported Rates attribute and the Supported Channel attribute, determine the maximum capability of the station. The columns are treated as logical AND when considering the highest rate reported for the station. Channel and frequency mappings are described in Section 5.1.4. If the same Device.WiFi.Event.STACapabilities.STAMAC submits association requests to both the 5 GHz and 2.4 GHz channel, use the 2.4 and 5 GHz rows in Table 8, Some stations were built with partial support for mandatory standard MCSs and most APs support multiple operations modes. If a station qualifies for two rows of the table, staMcsCapability should use the higher capacity classification. For example, 11ac instead of 11n for an AP that advertises 11g speeds at 2.4 GHz and 11ac capability rates at 5 GHz.

```
staMcsCapability = /* See Table 8 below*/
```

Table 8: Station Capability Classification

staMcsCapability	Rate = Max(Device.WiFi.Event.STACapabilities. SupportedBitrate)	Device.WiFi.Event.STACapabilities.ConnectFre quency
Legacy	Rate <= 2 Mbps	Only 2.4 GHz
11a	Rate <= 54 Mbps	Only 5 GHz
11b	Rate <= 11 Mbps	Only 2.4 GHz

staMcsCapability	Rate = Max(Device.WiFi.Event.STACapabilities. SupportedBitrate)	Device.WiFi.Event.STACapabilities.ConnectFre quency
11g	Rate <= 54 Mbps	Only 2.4 GHz
11n	500 Mbps > Rate > 54 Mbps	2.4 GHz and 5 GHz
11ac	1800 Mbps > Rate > 54 Mbps	5 GHz only/May report both for backward compatibility

5.1.5.2 Channel

Several data elements are available for determining the capabilities of the channel including:

```
apChanMcsCapability =
Device.WiFi.PhyCapa.Capabilities.ChannelNumber
Device.WiFi.PhyCapa.Capabilities.ChannelWidth
Device.WiFi.PhyCapa.Capabilities.MaxTxPower
Device.WiFi.PhyCapa.CurrentUTCTime
Device.WiFi.PhyCapa.HTMCS
Device.WiFi.PhyCapa.VHTRxMCSSet.NSS.MCS
Device.WiFi.PhyCapa.VHTTxMCSSet.NSS.MCS
Device.WiFi.PhyCapa.VHTShortGI
Device.WiFi.PhyCapa.Capabilities.Frequency
```

These capabilities could be useful in a number of contexts. For example, understanding the Device.WiFi.PhyCapa.Capabilities.MaxTxPower for a particular channel, and which channels are supported (Device.WiFi.PhyCapa.Capabilities.ChannelNumber, Device.WiFi.PhyCapa.Capabilities.Frequency) are important for channel selection algorithms, band steering algorithms, and power control algorithms. These are all key elements that could be configured by a Radio Resource Management (RRM) Virtual Network Function (VNF).

The larger the channel width, the higher burst capacities can be achieved over the air. The Device.WiFi.PhyCapa.Capabilities.ChannelWidth can be used with the RRM VNF to understand the options to set the channel size to the highest possible while accommodating other non-overlapping channels from other APs in the wireless environment.

Additionally, if older STAs with only 802.11b MCS capabilities exist on the network, disabling an MCS on some of the 2.4 GHz channels could be done to isolate the older devices to a specific channel, in order to avoid impacting performance of the other channels located nearby in the wireless environment. The MCS capabilities can be discovered from Device.WiFi.PhyCapa.HTMCS, Device.WiFi.PhyCapa.VHTRxMCSSet.NSS.MCS, Device.WiFi.PhyCapa.VHTTxMCSSet.NSS.MCS, and Device.WiFi.PhyCapa.VHTShortGI. The number of spatial streams can be useful in understanding the number of streams and beamforming capabilities of the AP.

By only detecting the presence of a VHT capability, you can determine if the channel is capable of 802.11ac v. only 802.11n bitrates. A VHT capability is also likely to have an HT capability, since most 802.11ac APs typically support 802.11n and earlier stations.

Appendix I Linux Wireless NL80211 Driver API

nl80211 is a driver API that connects user space applications, such as the agent sensor, to provide an interface between user space and kernel space in OpenWrt Linux, based on netlink libraries libnl or libnl-tiny. The elements of 80211 netlink⁴ interface are defined in the source code in nl80211.h.⁵

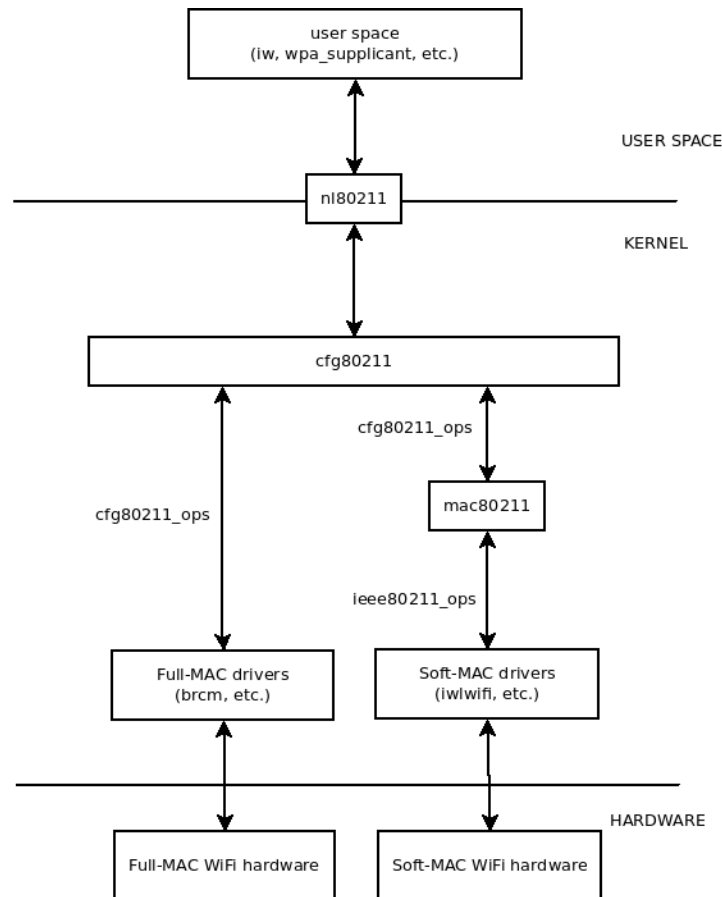


Figure 8: nl80211 Interface⁶

The nl80211 libraries organize the presentation of collected data into categories. The six categories of focus for the WCCF project focused on Wi-Fi include:

1. Survey
2. Station
3. Scan
4. Interface
5. PhyCapa
6. Event

⁴ <https://en.wikipedia.org/wiki/Netlink>

⁵ <http://lxr.free-electrons.com/source/include/uapi/linux/nl80211.h>

⁶ <https://i.stack.imgur.com/TNLK1.png>

Additional categories of metrics that could be developed in the future include those that describe router or AP functions including:

1. Interface Statistics for non-wireless interfaces such as Ethernet or MoCA
2. Process Statistics

General Attributes are attributes that are useful in more than one of the sensors.

I.1 General Attributes

Table 9: General Attributes

	Description	Format	Units	Note
NL80211_ATTR_IFINDEX	Network interface index of the device to operate on	String	NA	
NL80211_ATTR_IFNAME	Network interface name	String	NA	
NL80211_ATTR_MAC	MAC address (various uses)	6 octets	NA	
NL80211_RATE_INFO_BITRATE32	total bitrate	U32I	100 kbps	When getting information about the bitrate of a station. There are 2 attributes for bitrate – a legacy one that represents a 16-bit value, and new one that represents a 32-bit value. If the rate value fits into 16 bit, both attributes are reported with the same value. If the rate is too high to fit into 16 bits (>6.5535 Gbps) only 32-bit attribute is included. User space tools encouraged to use the 32-bit attribute and fallback to the 16-bit one for compatibility with older kernels
NL80211_RATE_INFO_MCS	MCS index for 802.11n	U8I		
NL80211_RATE_INFO_SHORT_GI	400ns guard interval	String		This may just be - if present, the SGI is used
NL80211_RATE_INFO_VHT_MCS	MCS index for 802.11ac	U8I		
NL80211_RATE_INFO_VHT_NSS	Number of streams in VHT	U8I		

I.2 Survey Collection Metrics

Table 10: Survey Metrics used in KPI Calculation

Metric	Description	Format	Units	Note
NL80211_SURVEY_INFO_FREQUENCY	Center Frequency of a Channel		MHz	
NL80211_SURVEY_INFO_IN_USE	Channel is currently being used	String		If Present
NL80211_SURVEY_INFO_NOISE	Noise level of channel	U8I	dBm	
NL80211_SURVEY_INFO_CHANNEL_TIME	Amount of time (in msec) that the radio was turned on (on channel or globally)	U32I	msec	
NL80211_SURVEY_INFO_CHANNEL_TIME_BUSY	Amount of the time the primary channel was sensed busy (either due to activity or energy detect)	U32I	msec	
NL80211_SURVEY_INFO_CHANNEL_TIME_EXT_BUSY	Amount of time the extension channel was sensed busy	U32I	msec	
NL80211_SURVEY_INFO_CHANNEL_TIME_RX	Amount of time the radio spent receiving data (on channel or globally)	U32I	msec	

Metric	Description	Format	Units	Note
NL80211_SURVEY_INFO_CHANNEL_TIME_TX	Amount of time the radio spent transmitting data (on channel or globally)	U32I	msec	

I.3 Station Collection Metrics

Table 11: STA Metrics used in KPI Calculation

Metric	Description	Format	Units	Note
NL80211_ATTR_RX_SIGNAL_DBM	Signal strength	U32I	dBm	
NL80211_ATTR_STA_SUPPORTED_CHANNELS	Array of supported channels			
NL80211_ATTR_STA_SUPPORTED_RATES	Supported rates, array of supported rates as defined by IEEE 802.11 7.3.2.2 but without the length restriction (at most NL80211_MAX_SUPP_RATES)			Max supported rates defined as 32 so this may return MCS indexes
NL80211_STA_INFO	Station information used when obtaining information about a station, specific attributes follow			
NL80211_STA_INFO_BEACON_SIGNAL_AVG	Signal strength average	U8I	dBm	Comment in patch says average is an exponential weighted moving average with a factor of 1024 and weight of 8
NL80211_STA_INFO_CHAIN_SIGNAL	Per receive chain signal strength of last PDU, contains a nested array of signal strength attributes	U8I	dBm	
NL80211_STA_INFO_INACTIVE_TIME	Time since last activity	U32I	msec	
NL80211_STA_INFO_RX_BYTES64	Total received bytes MPDU length from this station	U64I		
NL80211_STA_INFO_RX_PACKETS	Total received packet (MSDUs and MMPDUs) from this station	U32I		
NL80211_STA_INFO_SIGNAL	Signal strength of last received PPDU	U8I	dBm	
NL80211_STA_INFO_SIGNAL_AVG	Signal strength average	U8I	dBm	Comment in patch says average is an exponential weighted moving average with a factor of 1024 and weight of 8
NL80211_STA_INFO_TX_BYTES64	Total transmitted bytes MPDU length to this station	U64I		
NL80211_STA_INFO_TX_FAILED	Total failed packets (MPDUs) to this station	U32I		
NL80211_STA_INFO_TX_PACKETS	Total transmitted packet (MSDUs and MMPDUs) to this station	U32I		
NL80211_STA_INFO_TX_RETRIES	Total retries (MPDUs) to this station	U32I		

I.4 Scan Collection Metrics

Table 12: Scan Metrics used in KPI Calculation

Metric	Description	Format	Units	Note
NL80211_ATTR_SSID	SSID (binary attribute, 0..32 octets)			
NL80211_BSS_BSSID	BSSID of the BSS	6 octets		
NL80211_BSS_FREQUENCY	Frequency in MHz	U32I	MHz	
NL80211_BSS_SIGNAL_MBM	Signal strength of probe response/beacon in mBm	S32I	100*dBm	
NL80211_BSS_SIGNAL_UNSPEC	Signal strength of the probe response/beacon in unspecified units, scaled to 0..100	U8I		

I.5 Interfaces Collection Metrics

Table 13: Interface Metrics used in KPI Calculation

Metric	Description	Format	Units	Note
NL80211_ATTR_CENTER_FREQ1	Center frequency of the first part of the channel, used for anything but 20 MHz bandwidth	U32I		This is from nl80211_channel_width
NL80211_ATTR_CENTER_FREQ2	Center frequency of the second part of the channel, used only for 80+80 MHz bandwidth	U32I		This is from nl80211_channel_width
NL80211_CHAN_20_NO_HT	20 MHz, non-HT channel	String		this is from nl80211_channel_type
NL80211_CHAN_WIDTH_10	10 MHz OFDM channel	U32I		This is from nl80211_channel_width
NL80211_CHAN_WIDTH_160	160 MHz channel, the NL80211_ATTR_CENTER_FREQ1 attribute must be provided as well	U32I		This is from nl80211_channel_width
NL80211_CHAN_WIDTH_20	20 MHz HT channel	U32I		This is from nl80211_channel_width
NL80211_CHAN_WIDTH_20_NOHT	20 MHz, non-HT channel	U32I		This is from nl80211_channel_width
NL80211_CHAN_WIDTH_40	40 MHz channel, the NL80211_ATTR_CENTER_FREQ1 attribute must be provided as well	U32I		This is from nl80211_channel_width
NL80211_CHAN_WIDTH_5	5 MHz OFDM channel	U32I		This is from nl80211_channel_width
NL80211_CHAN_WIDTH_80	80 MHz channel, the NL80211_ATTR_CENTER_FREQ1 attribute must be provided as well	U32I		This is from nl80211_channel_width
NL80211_CHAN_WIDTH_80P80	80+80 MHz channel, the NL80211_ATTR_CENTER_FREQ1 and NL80211_ATTR_CENTER_FREQ2 attributes must be provided as well	U32I		This is from nl80211_channel_width
NL80211_CHAN_20_NO_HT	20 MHz, non-HT channel	String		This is from nl80211_channel_type
NL80211_CHAN_HT20	20 MHz HT channel	String		If present, this is from nl80211_channel_type
NL80211_CHAN_HT40MINUS	HT40 channel, secondary channel below the control channel	String		If present, this is from nl80211_channel_type
NL80211_CHAN_HT40PLUS	HT40 channel, secondary channel above the control channel	String		If present, this is from nl80211_channel_type
NL80211_CHAN_NO_HT	20 MHz, non-HT channel	String		This is from nl80211_channel_type

I.6 PhyCapa Collection Metrics

Table 14: PHY Capabilities Metrics used in KPI Calculation

Metric	Description	Format	Units	Note
NL80211_FREQUENCY_ATTR_FREQ	Center Frequency of Supported Channels		MHz	
NL80211_FREQUENCY_ATTR_NO_20_MHZ, NL80211_FREQUENCY_ATTR_NO_HT40_MINUS, NL80211_FREQUENCY_ATTR_NO_HT40_PLUS, NL80211_FREQUENCY_ATTR_NO_80_MHZ, NL80211_FREQUENCY_ATTR_NO_160_MHZ	These are the attributes that set the channel width capabilities of the WLAN channel on the AP	Strings as shown in column 1 translated to 20 MHz HT40- HT40+ 80 MHz 160 MHz by WCCF agent sensor	String in MHz	
NL80211_FREQUENCY_ATTR_MAX_TX_POWER	This is the maximum transmit power supported by the channel		dBm	
NL80211_BAND_ATTR_HT_MCS_SET	This is an array of MCS indexes supported by the channel		Array of Integer after processing	
NL80211_BAND_ATTR_VHT_CAPA, NL80211_BAND_ATTR_VHT_MCS_SET	These identify the number of spatial streams, short GI, and a list of MCS			

I.7 Event Collection Metrics

The event collection metrics are obtained from decoding the association requests from stations when they occur. See the specification for data elements. These operate through a different interface than the other sensors.

I.8 Process Collection Metrics

Table 15: Process Metrics used in KPI Calculation

Metric	Description	Format	Units	Note

Appendix II **ACKNOWLEDGEMENTS**

We wish to thank the following CableLabs participants contributing directly to this document:

Robert Cruickshank	PNM consultant
David Early	CableLabs consultant – Applied Broadband
Belal Hamzeh	CableLabs VP Wireless R&D
Kevin Kershaw	CableLabs Director SW R&D
Curtis Knittle	CableLabs VP Wired R&D
Jeff Pedigo	CableLabs consultant – Applied Broadband
Josh Redmore	CableLabs Project Lead
Bao Phommantha	CableLabs Project Manager
Dan Rice	CableLabs consultant – Applied Broadband
Jason Schnitzer	CableLabs consultant – Applied Broadband
John Thompson	CableLabs consultant – Applied Broadband
Joan Branham	CableLabs Senior Technical Editor
