WLAN Toolbox™ Release Notes

MATLAB®

MathWorks®

How to Contact MathWorks

	Latest news:	www.mathworks.com
	Sales and services:	www.mathworks.com/sales_and_services
	User community:	www.mathworks.com/matlabcentral
	Technical support:	www.mathworks.com/support/contact_us
	Phone:	508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Contents

R2021a

R2020b

R2020a

R2019b

R2019a

R2018b

R2023a

Version: 3.6

New Features

Bug Fixes

Compatibility Considerations

IEEE 802.11be trigger-based waveform generation and simulation

Configure and generate extremely high-throughput trigger-based (EHT TB) waveforms, as defined in IEEE® P802.11be™/D2.0, by using the `wlanWaveformGenerator` function with the `wlanEHTTBConfig` object.

Use the following functions to model the reception of EHT TB waveforms.

- The `wlanEHTDemodulate` function enables you to demodulate EHT TB time-domain waveforms.
- The `wlanEHTOFDMInfo` function enables you to get OFDM information for EHT TB configurations.
- The `wlanEHTDataBitRecover` function enables you to recover bits from the EHT-Data field of a demodulated EHT TB waveform.
- The `wlanEHTLTFChannelEstimate` function enables you to perform channel estimation using the extremely high-throughput long training field (EHT-LTF) of an EHT TB waveform.

The following functions now support the EHT TB format:

- `wlanFormatDetect`
- `wlanFieldIndices`
- `wlanSampleRate`

The "802.11be Packet Error Rate Simulation for Uplink Trigger-Based Format" example shows how to measure the packet error rate of the EHT TB format.

The updated "802.11be Waveform Generation" example now shows how to configure and generate EHT TB waveforms.

Detect human presence using WLAN signals and deep learning

The "Detect Human Presence Using WLAN Signals and Deep Learning" example shows how to use a convolutional neural network to detect human presence by using the channel state information in wireless local area networks.

IEEE 802.11be multi-user signal reception

Use the following functions to model the reception of extremely high-throughput multi-user (EHT MU) waveforms, as defined in IEEE P802.11be/D2.0.

- The `wlanEHTDemodulate` function enables you to demodulate EHT MU time-domain waveforms.
- The `wlanEHTOFDMInfo` function enables you to get OFDM information for EHT MU configurations.
- The `wlanEHTDataBitRecover` function enables you to recover bits from the EHT-Data field of a demodulated EHT MU waveform.
- The `wlanEHTLTFChannelEstimate` function enables you to perform channel estimation using the EHT-LTF of an EHT MU waveform.
- The `wlanEHTTrackPilotError` enables you to track and correct pilot errors in EHT MU waveforms.

The following functions now support the EHT MU format:

- `wlanFormatDetect`
- `wlanFieldIndices`
- `wlanReferenceSymbols`
- `wlanClosestReferenceSymbol`

WLAN system-level simulation

The `wlanNode` object and the associated object functions enable you to create and configure a WLAN node. Configure device settings using the `wlanDeviceConfig` object. To simulate wireless local area networks, use these objects together with the `wirelessNetworkSimulator`, `networkTrafficOnOff`, `networkTrafficVoIP`, `networkTrafficVideoConference`, and `networkTrafficFTP` objects from the Communications Toolbox™ Wireless Network Simulation Library add-on. These features enable you to:

- Assign roles like access point, station, or mesh to the node.
- Configure traffic between the nodes.
- Configure the connection between access point and station.
- Configure routing paths in mesh topology.
- Model multi-band networks.
- Model downlink OFDMA transmissions.
- Model non-SRG spatial reuse (BSS coloring).
- Switch between the abstracted models and full models of the MAC and PHY.

WLAN Waveform Generator app support in MATLAB Online

You can now use the **WLAN Waveform Generator** app in MATLAB® Online™.

Packet recovery and analysis updates

The "Recover and Analyze Packets in 802.11 Waveform" example now enables you to use software-defined radios supported by "Wireless Testbench"™.

Signal-to-noise ratio definition

The "SNR Definition in End-to-End Simulations" example shows how the signal-to-noise ratio (SNR) is defined in WLAN Toolbox end-to-end simulations.

Noise estimation

The `wlanLLTFNoiseEstimate` function enables you to estimate the mean noise power of a waveform using the legacy long training field (L-LTF) or S1G long training field (S1G-LTF).

Network simulation functionality moved to Communications Toolbox Wireless Network Simulation Library

These features have moved from WLAN Toolbox to the Communications Toolbox Wireless Network Simulation Library add-on. To use these features, download and install the add-on.

- networkTrafficOnOff
- networkTrafficVoIP
- networkTrafficVideoConference
- networkTrafficFTP

VHT channel estimation at pilot subcarrier locations

The wlanVHTLTFChannelEstimate function now optionally returns a channel estimate at the pilot subcarrier locations.

Single-precision support

The following functions now support single-precision values for their numeric input arguments.

- wlanFormatDetect
- wlanHESIGABitRecover
- wlanLSIGBitRecover
- wlanNonHTOFDMDemodulate
- wlanBCCDecode
- wlanBCCDeinterleave
- wlanConstellationDemap
- wlanVHTLTFChannelEstimate
- wlanPreHEChannelEstimate

Functionality being removed or changed

Enumeration classes in IEEE 802.11be waveform generation
Behavior change

- The SpatialMapping property of a wlanEHTRU object is now saved as a member of the wlan.type.SpatialMapping enumeration class.
- The ChannelCoding and PostFECPaddingSource properties of a wlanEHTUser object are now saved as members of the wlan.type.ChannelCoding and wlan.type.PostFECPaddingSource enumeration classes, respectively.

R2022b

Version: 3.5

New Features

Bug Fixes

Compatibility Considerations

IEEE 802.11be OFDMA and non-OFDMA waveform generation

Configure and generate OFDMA and non-OFDMA extremely high throughput multi-user (EHT MU) waveforms, as defined in IEEE P802.11be/D1.5., by using the `wlanWaveformGenerator` function with the `wlanEHTMUConfig` object. Configure resource unit and user data by using the `wlanEHTRU` and `wlanEHTUser` objects.

Wireless network simulator

Simulate a multinode wireless network by using the `wirelessNetworkSimulator` object and the associated object functions. Use this object to perform these tasks.

- Run the nodes and distribute the packets.
- Add nodes to the simulation.
- Perform scheduled actions.
- Perform channel modeling.

IEEE 802.11be downlink throughput example

Compare the MU-MIMO and OFDMA throughputs of an IEEE 802.11be downlink transmission in the 802.11be Downlink Multi-User MIMO and OFDMA Throughput Simulation example.

OFDM beacon receiver using software-defined radio example

The OFDM Beacon Receiver Using Software-Defined Radio example shows how to retrieve information about WiFi networks on the 5 GHz band using a software-defined radio (SDR).

IEEE 802.11ax channel estimation

Perform channel estimation on IEEE 802.11ax™ transmissions. Call the `wlanHELTFChannelEstimate` function to estimate the channel by using the high-efficiency Long Training field (HE-LTF). Call the `wlanPreHEChannelEstimate` function to estimate using legacy fields.

Single-precision support for demodulation and channel estimate functions

The `wlanHEDemodulate`, `wlanLLTFDemodulate`, and `wlanLLTFChannelEstimate` functions now support single-precision values for their numeric input arguments.

WLAN Waveform Generator App Updates

The WLAN Waveform Generator app now enables you to transmit your waveform over the air at full radio device rate. For a list of radios that support full device rates, see Supported Radio Devices (Wireless Testbench). This feature requires Wireless Testbench. For an example, see Transmit App-Generated Wireless Waveform Using Radio Transmitters.

Pilot amplitude tracking in HT, Non-HT, and VHT data recovery functions

The `wlanHTDataRecover`, `wlanNonHTDataRecover`, and `wlanVHTDataRecover` functions can now track and correct an average amplitude error with respect to the estimated received pilots.

Packet transmission time

Obtain the packet transmission time from transmission configuration objects by using the `transmitTime` object function.

Functionality being removed or changed

OFDM information functions now output sample rate
Behavior change

The output structures of these functions now include `SampleRate`, a field containing the sample rate of the associated waveform.

- `wlanDMGOFDMInfo`
- `wlanHEOFDMInfo`
- `wlanHTOFDMInfo`
- `wlanNonHTOFDMInfo`
- `wlanS1GOFDMInfo`
- `wlanVHTOFDMInfo`

R2022a

Version: 3.4

New Features

Bug Fixes

IEEE 802.11be Waveform Generation and Analysis Updates

The 802.11be Waveform Generation example now provides these capabilities, as defined in IEEE P802.11be/D1.2.

- Generate and decode extremely-high-throughput (EHT) waveforms in EHT duplicate (EHT DUP) mode, corresponding to modulation and coding scheme (MCS) index 14, as well as dual-carrier modulation binary phase-shift keying (DCM-BPSK), corresponding to MCS 15.
- Encode and decode the U-SIG field.
- Encode the EHT-SIG field.

The 802.11be Transmitter Measurements example now shows how to measure the transmitter modulation accuracy, spectral mask, and spectral flatness for IEEE 802.11be orthogonal frequency-division multiple access (OFDMA) waveforms.

MAC Frame Generation and Decoding Updates

You can now generate and decode medium access control (MAC) frames compatible with IEEE Std 802.11ax-2021 and MAC mesh data frames. Generate MAC frames by using the `wlanMACFrame` function with the updated `wlanMACFrameConfig`, `wlanMACTriggerConfig`, and `wlanMACTriggerUserConfig` objects. Decode MAC frames by using the updated `wlanMPDUDecode` function.

System-Level Simulation Examples

The 802.11ax Downlink OFDMA Multinode System-Level Simulation example shows how to model a WLAN multinode downlink orthogonal frequency-division multiple access (OFDMA) network consisting of an access point and four stations (STAs). You can select STAs for each transmission by using a round-robin or custom scheduling algorithm.

The Get Started with WLAN System-Level Simulation in MATLAB example now shows how to simulate IEEE 802.11™ mesh networks.

Noncollaborative Coexistence Between Bluetooth and WLAN Signals

The PHY Simulation of Bluetooth BR/EDR, LE, and WLAN Coexistence shows how to simulate homogeneous and heterogeneous coexistence between Bluetooth® basic rate/enhanced data rate (BR/EDR), low energy (LE), and WLAN signals by using physical layer (PHY) modeling. You can perform adaptive frequency hopping (AFH) by classifying the channels, and compute the bit error rate (BER) and signal-to-interference-plus-noise ratio (SINR).

Spectral Flatness Test Function

Test the spectral flatness of WLAN waveforms by using the `wlanSpectralFlatness` function. The function measures the spectral flatness and compares the measurement to the spectral flatness range specified in the appropriate IEEE 802.11 standard. The function supports these standards and corresponding formats.

Standard	Formats
IEEE P802.11be/D1.2	Extremely high throughput (EHT)
IEEE Std 802.11ax-2021	High-efficiency single-user (HE SU), HE extended-range single-user (HE ER SU), HE multi-user (HE MU), and HE trigger-based (HE TB)
IEEE Std 802.11-2020	Very high throughput (VHT), high throughput (HT), non-high-throughput (non-HT), and sub 1 GHz (S1G)

Channel Center Frequency Calculation

Calculate the WLAN channel center frequency in the 2.4, 5, and 6 GHz operating bands by using the `wlanChannelFrequency` function.

For information about IEEE 802.11 channel designations, see WLAN Radio Frequency Channels.

IEEE 802.11ax Waveform Pilot Error Tracking

Track and correct common phase error and amplitude error in high-efficiency (HE) waveforms by using the `wlanHETrackPilotError` function.

Capture and Decode Packets Using SDR and Preamble Detection

The OFDM WiFi Scanner Using SDR Preamble Detection example shows how to retrieve information about WLAN networks by capturing and decoding OFDM packets using a software-defined radio (SDR) and preamble detection.

WLAN Waveform Generator App Updates

The **WLAN Waveform Generator** app now provides these capabilities.

- Generate WLAN waveforms with fast-Fourier-transform-based (FFT-based) oversampling.
- Visualize the complementary cumulative distribution function (CCDF) of a WLAN waveform. Use the CCDF to measure the peak-to-average-power ratio (PAPR) of the waveform. You can also visualize the CCDF of signal bursts by selecting **Burst mode** in the **CCDF** tab of the app.

WLAN Stream Parser Updates

The `wlanStreamParse` and `wlanStreamDeparse` functions now support stream parsing and deparsing parameter sets compatible with IEEE P802.11be/D1.2. These functions also support single-precision values for their numeric input arguments.

Characteristic Information for TGay Channel Model

You can now obtain the path delay, Doppler shift, and angles of arrival and departure of rays traveling between transmit and receive antenna arrays in a TGay multipath fading channel model. Configure a

TGay channel model by using the `wlanTGayChannel` object and return its characteristic information by using the `info` object function.

R2021b

Version: 3.3

New Features

Bug Fixes

IEEE 802.11be Support

WLAN Toolbox functions now support the extremely high throughput (EHT) packet format, as defined in IEEE P802.11be/D1.0.

These functions support a channel bandwidth of 320 MHz.

Function	Description
wlanSampleRate	Nominal sample rate
wlanTGaxChannel	Filter signal through 802.11ax multipath channel
wlanCoarseCFOEstimate	Coarse carrier frequency offset (CFO) estimation
wlanFineCFOEstimate	Fine CFO estimation
wlanLLTFChannelEstimate	Legacy long training field (L-LTF) channel estimation
wlanLLTFDemodulate	L-LTF demodulation
wlanPacketDetect	OFDM packet timing offset estimation
wlanSymbolTimingEstimate	L-LTF fine symbol timing estimate

These functions support 4096-point quadrature amplitude modulation (4096-QAM).

Function	Description
wlanConstellationMap	Constellation mapping
wlanConstellationDemap	Constellation demapping
wlanReferenceSymbols	Find reference symbols of constellation diagram
wlanClosestReferenceSymbol	Find closest constellation points

The 802.11be Waveform Generation example shows how to parameterize and generate IEEE 802.11be EHT multi-user (EHT-MU) waveforms.

The 802.11be Packet Error Rate Simulation for an EHT MU Single-User Packet Format example shows how to measure the packet error rate of an IEEE 802.11be EHT single-user link.

IEEE 802.11ba Waveform Generation and Analysis

Configure and generate wake-up radio (WUR) waveforms, as defined in IEEE P802.11ba™/D8.0, by using the wlanWaveformGenerator function with the wlanWURConfig and wlanWURSubchannel objects.

These functions now support the WUR packet format.

Function	Description
getPSDULength	PSDU length
wlanFieldIndices	PPDU field indices
wlanSampleRate	Nominal sample rate

The 802.11ba WUR Waveform Generation and Analysis example shows how to generate WUR waveforms and measure their transmit spectrum mask and spectral flatness.

System-Level Simulation of IEEE 802.11ax Networks

The 802.11ax Multinode System-Level Simulation of Residential Scenario Using MATLAB example now supports abstracted and full models of the physical layer (PHY) and medium access control (MAC) layer. You can generate abstracted and full MAC frames and switch between abstracted and full models of the PHY.

The Get Started with WLAN System-Level Simulation in MATLAB example models an IEEE 802.11ax network by using abstracted and full models of the PHY and MAC layer. You can simulate a multinode WLAN network with abstracted or full MAC frames and abstracted or full models of the PHY.

Waveform Generation with Oversampling

These functions now support waveform generation with fast-Fourier-transform-based (FFT-based) oversampling. To specify an oversampling factor in any of these functions, call the function with the `OversamplingFactor` name-value argument.

Function	Description
wlanWaveformGenerator	WLAN waveform
wlanVHTData	Very high throughput Data (VHT-Data) field waveform
wlanVHTLTF	VHT long training field (VHT-LTF) waveform
wlanVHTSIGA	VHT SIGNAL A (VHT-SIG-A) field waveform
wlanVHTSIGB	VHT-SIG-B field waveform
wlanVHTSTF	VHT short training field (VHT-STF) waveform
wlanHTData	HT-Data field waveform
wlanHTSIG	HT-SIG field waveform
wlanHTSTF	HT-STF waveform
wlanLLTF	Legacy long training field (L-LTF) waveform
wlanLSIG	L-SIG field waveform
wlanLSTF	L-STF waveform
wlanNonHTData	Non-HT-Data field waveform
wlanFieldIndices	PPDU field indices
wlanSampleRate	Nominal sample rate
wlanHEOFDMInfo	High-efficiency (HE) orthogonal frequency-division multiplexing (OFDM) information
wlanS1GOFDMInfo	Sub-1-GHz (S1G) OFDM information
wlanVHTOFDMInfo	VHT OFDM information
wlanHTOFDMInfo	HT OFDM information
wlanNonHTOFDMInfo	Non-HT OFDM information

Image Transmission and Reception with IEEE 802.11 Waveforms and SDR

The Image Transmission and Reception Using 802.11 Waveform and SDR example shows how to transmit and receive images that are encoded in IEEE 802.11 waveforms. You can configure the example to send waveforms through an additive white Gaussian noise (AWGN) channel or transmit and receive waveforms over the air by using a software-defined radio (SDR).

WLAN Waveform Generator App Updates

The **WLAN Waveform Generator** app now supports the configuration, generation, and visualization of 80 MHz and 160 MHz high-efficiency multi-user (HE MU) transmissions with punctured 20 MHz subchannels. You can configure the primary 20 MHz subchannel and puncture other 20 MHz subchannels for efficient coexistence with legacy IEEE 802.11 networks.

The **WLAN Waveform Generator** now enables you to export WLAN waveforms to Simulink®. Generate a waveform in a Simulink model by using the Waveform From Wireless Waveform Generator App block. For more information, see the Generate Wireless Waveform in Simulink Using App-Generated Block tutorial.

MATLAB Online Support

MATLAB Online now supports WLAN Toolbox software. For more information about supported toolboxes, see Specifications and Limitations.

R2021a

Version: 3.2

New Features

Bug Fixes

Compatibility Considerations

IEEE 802.11be Single-User Support

The 802.11be 4096-QAM 320 MHz Waveform Generation and Analysis example shows how to generate an IEEE 802.11be single-user waveform with a channel bandwidth of 320 MHz and 4096-point quadrature amplitude modulation (4096-QAM). The example shows how to measure transmitter modulation accuracy, spectrum emission mask, and spectral flatness.

IEEE 802.11ax HE TB Feedback NDP Support

Configure and generate high-efficiency trigger-based (HE TB) feedback null data packets (NDPs) by using the `wlanWaveformGenerator` function with the updated `wlanHETBConfig` object and its associated `getNDPFeedbackConfiguration` object function.

Demodulate received HE TB feedback NDPs by using the `wlanHEDemodulate` and `wlanHEOFDMInfo` functions with the `wlanHETBConfig` object. Recover NDP feedback status by using the `wlanHETBNDPFeedbackStatus` function.

The 802.11ax Feedback Status Misdetection Simulation for Uplink Trigger-Based Feedback NDP example shows how to measure the probability of incorrectly detecting feedback status information in an uplink HE TB feedback NDP transmission from multiple stations (STAs).

IEEE 802.11az Positioning and Localization

The Three-Dimensional Indoor Positioning with 802.11az Fingerprinting and Deep Learning example shows how to train a convolutional neural network (CNN) for localization and positioning by using Deep Learning Toolbox™ features and IEEE 802.11az™ data generated using WLAN Toolbox features.

The updated 802.11az Positioning Using Super-Resolution Time of Arrival Estimation example shows how to estimate the position of a STA in a multipath environment by using a multiple signal classification (MUSIC) algorithm defined in the IEEE 802.11az standard.

WLAN Waveform Generator App Updates

The **WLAN Waveform Generator** app now supports the configuration, generation, and visualization of IEEE 802.11 non-high-throughput (non-HT) duplicate packets.

Generate and Parse IEEE 802.11 MAC Trigger Frames

Generate WLAN medium access control (MAC) trigger frames by using the updated `wlanMACFrame` function with the `wlanMACFrameConfig`, `wlanMACTriggerConfig`, and `wlanMACTriggerUserConfig` objects.

Decode WLAN MAC trigger frames by using the updated `wlanMPDUDecode` function with the `wlanMACFrameConfig`, `wlanMACTriggerConfig`, and `wlanMACTriggerUserConfig` objects.

You can generate and decode basic, multi-user block ack request (MU-BAR), and multi-user request to send (MU-RTS) MAC trigger frames, as defined in IEEE P802.11ax/D4.1.

Write MAC Frames to PCAP or PCAPNG File

Write generated and recovered WLAN MAC frames to a file with the `.pcap` or `.pcapng` extension by using the `wlanPCAPWriter` object and its associated `write` object function.

Application Traffic Pattern Generation

Generate a video conference application traffic pattern by using the `networkTrafficVideoConference` object and its associated `generate` object function. Accurately model real-world data traffic by using this object in system-level simulations.

Model and Test IEEE 802.11ax RF Receiver with 5G interference

The Modeling and Testing an 802.11ax RF Receiver with 5G Interference example shows how to characterize the impact of RF impairments, such as phase noise and power amplifier nonlinearities, on IEEE 802.11ax waveform reception with an adjacent 5G or 802.11ax interferer. The example shows how to generate baseband waveforms by using WLAN Toolbox and 5G Toolbox™ features and how to model the RF receiver by using RF Blockset™ features.

Model IEEE 802.11ax Residential Scenario

The 802.11ax Multinode System-Level Simulation of Residential Scenario Using MATLAB example shows how to configure and simulate a multinode IEEE 802.11ax residential scenario and how to measure network statistics such as throughput, latency, and packet loss.

Measure IQ Gain and Phase Imbalance of IEEE 802.11ac Waveform

The 802.11ac Transmitter Measurements example now shows how to measure the in-phase and quadrature (IQ) gain and phase imbalance of an IEEE 802.11ac™ waveform.

Single-Precision Support

These functions now support single-precision values for their numeric input arguments.

Function	Description	Single-Precision Support
`wlanCoarseCFOEstimate`	Perform coarse carrier frequency offset (CFO) estimation	All numeric input arguments
`wlanFineCFOEstimate`	Perform fine CFO estimation	All numeric input arguments
`wlanPacketDetect`	Detect orthogonal frequency-division multiplexing (OFDM) packet by using legacy short training field (L-STF)	`rxSig` input only
`wlanSymbolTimingEstimate`	Perform fine symbol timing estimation by using legacy long training field (L-LTF)	`rxSig` input only

Generate IEEE 802.11ax Waveforms with Punctured 20 MHz Subchannels

Configure 80 MHz and 160 MHz high-efficiency multi-user (HE MU) transmissions with punctured 20 MHz subchannels by setting the `PrimarySubchannel` and `AllocationIndex` properties of the updated `wlanHEMUConfig` object. You can configure the primary 20 MHz subchannel and puncture other 20 MHz subchannels for efficient coexistence with legacy IEEE 802.11 networks.

Functionality Being Removed or Changed

wlanRecoveryConfig has been removed
Errors

The `wlanRecoveryConfig` object has been removed.

Starting in R2020a, all functions that supported the `wlanRecoveryConfig` object as an input argument support name-value arguments corresponding to the properties of the object. To parameterize an affected function, use one or more of these name-value arguments instead of the `wlanRecoveryConfig` object as an input argument. This table lists the affected functions, their syntaxes that are no longer supported, and the recommended replacement syntax for each case.

Function	Removed Syntax	Recommended Replacement Syntax
wlanFormatDetect	`wlanFormatDetect(rxSig,chEst,noiseVarEst,cbw,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanFormatDetect(rxSig,chEst,noiseVarEst,cbw,Name,Value)`, where `Name,Value` specifies one or more name-value arguments
wlanHTDataRecover	`wlanHTDataRecover(rxSig,chEst,noiseVarEst,cfg,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanHTDataRecover(rxSig,chEst,noiseVarEst,cfg,Name,Value)`, where `Name,Value` specifies one or more name-value arguments
wlanHTSIGRecover	`wlanHTSIGRecover(rxSig,chEst,noiseVarEst,cbw,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanHTSIGRecover(rxSig,chEst,noiseVarEst,cbw,Name,Value)`, where `Name,Value` specifies one or more name-value arguments
wlanLSIGRecover	`wlanLSIGRecover(rxSig,chEst,noiseVarEst,cbw,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanLSIGRecover(rxSig,chEst,noiseVarEst,cbw,Name,Value)`, where `Name,Value` specifies one or more name-value arguments
wlanNonHTDataRecover	`wlanNonHTDataRecover(rxSig,chEst,noiseVarEst,cfg,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanNonHTDataRecover(rxSig,chEst,noiseVarEst,cfg,Name,Value)`, where `Name,Value` specifies one or more name-value arguments

Function	Removed Syntax	Recommended Replacement Syntax
`wlanVHTDataRecover`	`wlanVHTDataRecover(__,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanVHTDataRecover(__,Name,Value)`, where `Name,Value` specifies one or more name-value arguments
`wlanVHTSIGARecover`	`wlanVHTSIGARecover(rxSig,chEst,noiseVarEst,cbw,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanVHTSIGARecover(rxSig,chEst,noiseVarEst,cbw,Name,Value)`, where `Name,Value` specifies one or more name-value arguments
`wlanVHTSIGBRecover`	`wlanVHTSIGBRecover(__,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanVHTSIGBRecover(__,Name,Value)`, where `Name,Value` specifies one or more name-value arguments

R2020b

Version: 3.1

New Features

Bug Fixes

IEEE 802.11az Draft 2.0 Support

The 802.11az Waveform Generation example shows how to parameterize and generate HE ranging null data packet (NDP) waveforms with optional secure long training fields (LTFs), as defined in IEEE P802.11az/D2.0.

The 802.11az Indoor Positioning Using Time of Arrival example shows how to estimate the position of an indoor device by using the time of arrival and trilateration technique.

Non-HT Duplicate Transmission and Reception

Configure and generate IEEE 802.11 non-high-throughput (non-HT) duplicate packets by using the `wlanWaveformGenerator` function with the updated `wlanNonHTConfig` object. You can also configure and generate the Data field of a non-HT duplicate transmission by using the `wlanNonHTData` function.

Recover data from non-HT duplicate transmissions by using these functions: `wlanFieldIndices`, `wlanInterpretScramblerState`, `wlanNonHTDataBitRecover`, `wlanNonHTDataRecover`, `wlanNonHTOFDMDemodulate`, and `wlanNonHTOFDMInfo`.

Spatial Reuse and BSS Coloring Support

The Spatial Reuse with BSS Coloring in 802.11ax Residential Scenario example shows how to perform spatial reuse with basic service set (BSS) coloring, as specified in IEEE P802.11ax/D4.1, in a system-level simulation. The example shows how to measure the throughput of each BSS and of the network by simulating overlapping BSS packet-detection-based spatial reuse operations.

Model and Test IEEE 802.11ax RF Transmitter

The Modeling and Testing an 802.11ax RF Transmitter example characterizes the impact of RF impairments, such as in-phase and quadrature imbalance, phase noise, and power amplifier nonlinearities, on IEEE 802.11ax waveform transmission. The example demonstrates how to measure the constellation, error vector magnitude (EVM), and spectrum attributes such as the mask, occupied bandwidth, channel power, and complementary cumulative distribution function (CCDF) by using WLAN Toolbox and RF Blockset features.

Write MAC Frames to PCAP File

Write generated and recovered WLAN medium access control (MAC) frames to a file with the `.pcap` or `.pcapng` extension by using the `pcapWriter` or `pcapngWriter` object, respectively, and the associated `write` object function.

With the `pcapWriter` object, you can configure the file name and add a timestamp to each packet.

With the `pcapngWriter` object, you can configure the file name, add a timestamp and metadata for each packet, and write custom data.

MAC Frame Generator and Parser Updates

Generate WLAN MAC frames with four address fields by using the `wlanMACFrame` function with the updated `wlanMACFrameConfig` object. Parse MAC frames with four address fields by using MAC frame parser functions `wlanAMPDUDeaggregate` and `wlanMPDUDecode`.

The MAC frame parser functions now support these additional IEEE 802.11 physical layer (PHY) configurations.

- Parse MAC frames in HE multi-user (HE MU) or HE trigger-based (HE TB) transmissions by using the parser functions with the `wlanHEMUConfig` or `wlanHETBConfig` object, respectively.

- Parse MAC frames recovered from HE single-user (HE SU), extended-range SU (HE ER SU), or HE MU transmissions by using the parser functions with the `wlanHERecoveryConfig` object.

Suppress all warnings when attempting to parse unsupported MAC frames by specifying the `'SuppressWarnings'` name-value pair argument of the parser functions. For a list of supported MAC frame formats, see the 802.11 MAC Frame Generation example.

Application Traffic Pattern Generation

Generate a file transfer protocol (FTP), a voice over internet protocol (VoIP), or an On-Off application traffic pattern by using the `networkTrafficFTP`, `networkTrafficVoIP`, or `networkTrafficOnOff` object, respectively, and the associated `generate` object function. You can use these objects in system-level simulations to accurately model real-world data traffic.

The Generate and Visualize FTP Application Traffic Pattern example shows how to generate an FTP application traffic pattern based on the IEEE 802.11ax Evaluation Methodology and the 3GPP TR 36.814 specification.

App-Based WLAN Waveform Generation

The **WLAN Waveform Generator** app now supports configuration, generation, and visualization of:

- HE TB waveforms, as defined in IEEE P802.11ax/D4.1

- All supported IEEE 802.11 waveforms with an arbitrary number of transmit antennas

The App-Based WLAN Waveform Generation example outlines waveform configuration, generation, and visualization workflows.

IEEE 802.11ax Post-FEC Padding

Generate HE waveforms with post-forward-error-correction (post-FEC) padding bits by using the `wlanWaveformGenerator` function with the updated `wlanHESUConfig`, `wlanHEMUConfig`, `wlanHEMUUser`, or `wlanHETBConfig` configuration object. To determine the number of required post-FEC padding bits for each format, use the `getNumPostFECPaddingBits` object function for the corresponding configuration object. You can configure the source of the post-FEC padding bits to be the current global random number stream, the mt19937ar algorithm, or custom post-FEC padding bits.

R2020a

Version: 3.0

New Features

Bug Fixes

Compatibility Considerations

IEEE 802.11ax Draft 4.1 Support

Generate high-efficiency single-user (HE SU) null data packets (NDPs) with preamble puncturing, as defined in IEEE P802.11ax/D4.1, by using the `wlanWaveformGenerator` function with the updated `wlanHESUConfig` object.

IEEE 802.11ax Trigger-Based Format

Configure and generate high-efficiency trigger-based (HE TB) waveforms by using the `wlanWaveformGenerator` function with the `wlanHETBConfig` object.

Demodulate and decode HE TB waveforms by using the `wlanHEDemodulate` and `wlanHEDataBitRecover` functions.

The updated 802.11ax Packet Error Rate Simulation for Uplink Trigger-Based Format example demonstrates minimum-mean-square-error-based ordered successive interference cancellation (MMSE-SIC) equalization.

Data Recovery Functions Using New Multicore LDPC Decoding Algorithms

Reduce simulation times by using low-density parity-check (LDPC) decoding with multicore processing in these data recovery functions: `wlanDMGDataBitRecover`, `wlanDMGHeaderBitRecover`, `wlanHEDataBitRecover`, `wlanHTDataRecover`, and `wlanVHTDataRecover`.

These functions support the layered belief propagation, normalized min-sum, and offset min-sum LDPC decoding algorithms.

Transmit and Receive Signals with Unlimited Antennas

Transmit and receive WLAN signals with an arbitrary number of antennas for any of these formats.

- HE SU
- HE extended range SU (HE ER SU)
- HE multi-user (HE MU)
- HE TB
- Very high throughput (VHT)
- High throughput (HT)
- Non-HT

Filter a signal through a multipath fading channel with an arbitrary number of links by using these updated channel models: `wlanTGacChannel`, `wlanTGahChannel`, `wlanTGaxChannel`, and `wlanTGnChannel`.

Blindly Detect, Decode, and Analyze Packets in Recovered Waveform

The Recover and Analyze Packets in 802.11ax Waveform example demonstrates how to detect, decode, and analyze 802.11a™ and 802.11ax packets in a recovered waveform. The receiver can

decode OFDM non-HT, HE MU, HE SU, and HE ER SU packets without knowledge of any transmission parameters aside from the channel bandwidth. You can configure the example to capture over-the-air waveforms by using software-defined radio (SDR) devices.

Multinode 802.11ax Downlink Throughput Comparison of OFDM and OFDMA

The 802.11ax Downlink Throughput Comparison of OFDM and OFDMA Through System-level Simulation example shows how to configure and simulate multinode downlink OFDM and OFDMA 802.11ax transmissions by using SimEvents®, Stateflow®, and WLAN Toolbox software.

Single-Precision Support in WLAN Channel Models

Reduce simulation times and memory consumption by specifying applicable properties of these channel models with single precision: `wlanTGacChannel`, `wlanTGahChannel`, `wlanTGaxChannel`, and `wlanTGnChannel`.

Wireless Waveform Generator App Update

Generate HE waveforms with packet extension by using the **Wireless Waveform Generator** app.

Functionality Being Removed or Changed

wlanRecoveryConfig will be removed
Warns

The `wlanRecoveryConfig` object will be removed in a future release.

Starting in R2020a, all functions that support a `wlanRecoveryConfig` object as an input argument now support name-value pair arguments corresponding to the properties of the object. To parameterize an affected function, use one or more of these name-value pairs instead of a `wlanRecoveryConfig` object. This table lists the affected functions, their syntaxes that are no longer recommended, and the recommended replacement syntax for each case.

Function	Not Recommended	Recommended
wlanFormatDetect	wlanFormatDetect(rxSig,chEst,noiseVarEst,cbw,cfgRec), where cfgRec is a wlanRecoveryConfig object	wlanFormatDetect(rxSig,chEst,noiseVarEst,cbw,Name,Value), where Name,Value specifies one or more name-value pair arguments
wlanHTDataRecover	wlanHTDataRecover(rxSig,chEst,noiseVarEst,cfg,cfgRec), where cfgRec is a wlanRecoveryConfig object	wlanHTDataRecover(rxSig,chEst,noiseVarEst,cfg,Name,Value), where Name,Value specifies one or more name-value pair arguments

Function	Not Recommended	Recommended
`wlanHTSIGRecover`	`wlanHTSIGRecover(rxSig,chEst,noiseVarEst,cbw,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanHTSIGRecover(rxSig,chEst,noiseVarEst,cbw,Name,Value)`, where `Name,Value` specifies one or more name-value pair arguments
`wlanLSIGRecover`	`wlanLSIGRecover(rxSig,chEst,noiseVarEst,cbw,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanLSIGRecover(rxSig,chEst,noiseVarEst,cbw,Name,Value)`, where `Name,Value` specifies one or more name-value pair arguments
`wlanNonHTDataRecover`	`wlanNonHTDataRecover(rxSig,chEst,noiseVarEst,cfg,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanNonHTDataRecover(rxSig,chEst,noiseVarEst,cfg,Name,Value)`, where `Name,Value` specifies one or more name-value pair arguments
`wlanVHTDataRecover`	`wlanVHTDataRecover(__,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanVHTDataRecover(__,Name,Value)`, where `Name,Value` specifies one or more name-value pair arguments
`wlanVHTSIGARecover`	`wlanVHTSIGARecover(rxSig,chEst,noiseVarEst,cbw,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanVHTSIGARecover(rxSig,chEst,noiseVarEst,cbw,Name,Value)`, where `Name,Value` specifies one or more name-value pair arguments
`wlanVHTSIGBRecover`	`wlanVHTSIGBRecover(__,cfgRec)`, where `cfgRec` is a `wlanRecoveryConfig` object	`wlanVHTSIGBRecover(__,Name,Value)`, where `Name,Value` specifies one or more name-value pair arguments

R2019b

Version: 2.2

New Features

IEEE 802.11ax Signal Recovery

Decode the HE-SIG-B common and user fields in a high-efficiency multiuser (HE MU) transmission by using the `wlanHESIGBCommonBitRecover` and `wlanHESIGBUserBitRecover` functions.

IEEE 802.11ax Packet Extension

Generate HE-format packets with packet extension, as specified in Draft 3.1 of the IEEE 802.11ax standard, by using the `wlanWaveformGenerator` function with the updated `wlanHESUConfig` and `wlanHEMUConfig` objects.

Detect Format of IEEE 802.11 Waveforms

The `wlanFormatDetect` function now detects these 802.11ax packet formats in a received waveform: HE single-user (HE SU), HE extended-range single-user (HE ER SU), HE multiuser (HE MU), and HE trigger-based (HE TB).

Calculate PSDU and APEP Length from Transmission Time

You can now generate single-user IEEE 802.11 waveforms for a given transmission duration or number of symbols.

- Calculate PSDU length by using the `wlanPSDULength` function.
- Calculate APEP length by using the `wlanAPEPLength` function.

System-Level Simulation of IEEE 802.11 Networks

These examples show how to simulate multinode 802.11 networks by using SimEvents, Stateflow, and WLAN Toolbox.

- 802.11 MAC and Application Throughput Measurement – Measure MAC and application layer throughput in an 802.11a/n/ac/ax network.
- 802.11 MAC QoS Traffic Scheduling – Simulate quality of service (QoS) traffic scheduling in an 802.11a/n/ac/ax network.
- 802.11ax System-Level Simulation with Physical Layer Abstraction and Physical Layer Abstraction for System-Level Simulation – Model a large 802.11ax network by using physical layer abstraction.

IEEE 802.11ax RU Visualization

Display the resource unit (RU) allocation for HE-format transmissions by using the `showAllocation` object function with the `wlanHESUConfig` and `wlanHEMUConfig` objects.

Wireless Waveform Generator App Updates

You can now export a WLAN waveform generated using the **Wireless Waveform Generator** app as a MATLAB script with a `.m` extension. You can recreate the waveform by running the exported script in MATLAB.

R2019a

Version: 2.1

New Features

IEEE 802.11ax Signal Recovery Functions: Decode L-SIG and HE-SIG-A fields

You can now decode the L-SIG and HE-SIG-A fields for IEEE 802.11ax format. The 802.11ax Signal Recovery with Preamble Decoding example shows how to recover single-user and multi-user 802.11ax signals by using the `wlanHERecoveryConfig` object and the `wlanLSIGBitRecover` and `wlanHESIGABitRecover` functions.

IEEE 802.11ay Support: Simulate 802.11ay propagation conditions

Use the `wlanTGayChannel` System object™ to simulate 802.11ay™ propagation conditions as demonstrated in the 802.11ad Packet Error Rate Single Carrier PHY Simulation with TGay Channel example. The ray-tracing-based channel model, described by Task Group ay, supports antenna array specification, SU-SISO and SU-MIMO, unpolarized and polarized signals, and beamforming. For more information, see WLAN Channel Models and Propagation Channel Models.

Wireless Waveform Generator App Updates

With the **Wireless Waveform Generator** app, you can now:

- Create WLAN waveforms that are compliant with 802.11ax modulation schemes.
- Generate a waveform that you can transmit with a connected lab test instrument. The Wireless Waveform Generator app can generate waveforms for instruments supported by the `rfsiggen` function. Use of the transmit feature in the Wireless Waveform Generator app requires Instrument Control Toolbox™.

MAC Frame Parser Functions: Deaggregate A-MPDUs and decode MPDUs

You can now parse WLAN medium access control (MAC) frames. Decode MAC protocol data units (MPDUs) with the `wlanMPDUDecode` function and deaggregate aggregated MPDUs (A-MPDUs) with the `wlanAMPDUDeaggregate` function.

This feature is demonstrated in the added 802.11 MAC Frame Decoding example and the updated 802.11 OFDM Beacon Receiver with Live Data example.

IEEE 802.11ax Draft 3.1 Support

With the added IEEE 802.11ax support, you can:

- Generate HE-format WLAN MAC frames compliant with IEEE 802.11ax Draft 3.1 using the `wlanMACFrame` function.
- Simulate an HE-MU packet with more than 16 HE-SIG-B symbols.
- Generate HE packets containing a PSDU of up to 6,500,631 bytes.

System-Level Simulation Examples: Simulate a WLAN network

These examples show you how to simulate 802.11 networks by simulating individual links between different nodes:

- 802.11ax PHY-Focused System-Level Simulation
- Multi-Node 802.11a Network Modeling with PHY and MAC

R2018b

Version: 2.0

New Features

Bug Fixes

IEEE 802.11ax Support: Generate IEEE 802.11ax (Draft 2.0) high-efficiency (HE) waveforms and simulate end-to-end links

WLAN Toolbox now supports 802.11ax Draft 2.0 waveform generation and link-level simulation of single user (HE-SU, HE-EXT-SU) and multi-user (HE-MU) formats.

The 802.11ax Packet Error Rate Simulation for Uplink Trigger-Based Format example shows how to measure the packet error rate of an uplink trigger-based (HE-TB) format link.

Wireless Waveform Generator App: Create, impair, visualize, and export modulated waveforms

Using the **Wireless Waveform Generator** app you can:

- Create OFDM, QAM, or PSK modulated waveforms or a sine wave test waveform.
- Create WLAN waveforms that are compliant with various 802.11 modulation schemes (a/b/g/j/p/n/ac/ad/ah).
- Add RF impairments to your waveforms.
- Plot waveforms in constellation diagram, spectrum analyzer, and time scopes.
- Export waveforms from the app to your MATLAB workspace or a file.

MAC Frame Support: Generate data, management, and control frames

This release adds support for inbuilt MAC Frame generation. This feature includes the following functions:

- `wlanMACFrame`: WLAN MAC frame generation (MPDU or A-MPDU)
- `wlanMSDULengths`: Returns a vector of MSDU lengths required to generate a MAC frame of given length, MAC configuration, and PHY configuration.
- `wlanMACFrameConfig`: MAC frame configuration object with properties relevant to the IEEE 802.11 MAC frame.
- `wlanMACManagementConfig`: MAC management configuration object with properties relevant to the IEEE 802.11 MAC management frame body.

IEEE TGay Channel Model Example: Simulate an end-to-end, DMG format WLAN link over a TGay millimeter wave channel

The 802.11ad Packet Error Rate Single Carrier PHY Simulation with TGay Channel example shows how to determine the packet error rate for an 802.11ad™ SC link with a TGay fading channel at a selection of SNR points.

IEEE 802.11ad RF Beamforming Example: Simulate an end-to-end, DMG format WLAN link with RF beamforming

The 802.11ad Single Carrier Link with RF Beamforming in Simulink example shows how to model an IEEE 802.11ad single carrier link with RF beamforming. This example also displays the packet error rate after processing each packet, the equalized symbol constellation, and EVM measurements.

IEEE 802.11p V2V Link Example: Simulate an end-to-end link over a vehicular channel model

The 802.11p Packet Error Rate Simulation for a Vehicular Channel example shows how to measure the packet error rate of an IEEE 802.11p™ link with a V2V fading channel and additive white Gaussian noise.

Functionality Being Removed or Changed

The `wlanGeneratorConfig` object has been removed. Use `Name, Value` pair syntax of `wlanWaveformGenerator` for parameterizing the `wlanWaveformGenerator` function. See `wlanWaveformGenerator` for the recommended parameter `Name,Value` pair syntax.

R2018a

Version: 1.5

New Features

IEEE 802.11ax Examples: Generate IEEE 802.11ax high efficiency (HE) waveforms and simulate downlink end-to-end links

This release adds three examples which show how WLAN System Toolbox functions can be used to simulate IEEE 802.11ax Draft 1.1:

- 802.11ax Parameterization for Waveform Generation and Simulation shows how to parameterize and generate different types of IEEE 802.11ax high efficiency (HE) formats.
- 802.11ax Packet Error Rate Simulation for Single User Format shows how to measure the packet error rate of an IEEE 802.11ax high efficiency (HE) single user format link.
- 802.11ax OFDMA and Multi-User MIMO Throughput Simulation shows the transmit and receive processing for an IEEE 802.11ax multi-user downlink transmission over a TGax indoor fading channel. Three transmission modes are simulated: OFDMA, MU-MIMO, and a combination of OFDMA and MU-MIMO.

IEEE TGax Indoor Channel Model: Simulate indoor 802.11ax propagation conditions for both single and multiuser scenario

In this release, WLAN System Toolbox™ provides the channel model described by the TGax task group. Use the wlanTGaxChannel System object to model 802.11ax indoor propagation conditions. For more information, see WLAN Channel Models and Propagation Channel.

IEEE 802.11ad Single Carrier PHY Example: Simulate an end-to-end, DMG format WLAN link over an AWGN with synchronization and equalization

This example enables you to study synchronization and equalization techniques in an end-to-end IEEE 802.11ad Single Carrier PHY link.

IEEE 802.11ad Extended MCS Support: Generate IEEE 802.11ad waveforms with the extended modulation and coding schemes (MCS) for the single carrier PHY

The toolbox now supports the extended modulation and coding schemes (MCS) for the IEEE 802.11ad single carrier PHY as specified in IEEE 802.11-2016.

R2017b

Version: 1.4

New Features

IEEE 802.11ad Receiver: Demodulate and decode directional multi-gigabit waveforms

This release adds support for IEEE 802.11ad demodulation and decoding steps. The 802.11ad specification defines the WLAN directional multi-gigabit (DMG) format.

This feature is contained in these functions:

- `wlanDMGDataBitRecover`: Recovers data bits from DMG Data field.
- `wlanDMGHeaderBitRecover`: Recovers header bits from DMG Header field.

Three new IEEE 802.11ad examples explain how to perform an end-to-end simulation in the different PHY types.

IEEE 802.11ah Example: Simulate an end-to-end sub 1 GHz link with the TGah channel model

802.11ah Packet Error Rate Simulation for 2x2 TGah Channel shows how to measure the packet error rate of an IEEE 802.11ah™ S1G short preamble link with a fading TGah indoor channel model and additive white Gaussian noise.

Simulink Example: Model a WLAN link in Simulink

WLAN Link Modeling in Simulink: Use the end-to-end 802.11n™ HT link example to explore the capabilities of WLAN System Toolbox in Simulink.

Bit-Level Processing Functions: Explore standard-compliant bit-level algorithms

WLAN Toolbox includes standard compliant bit-level functions:

- `wlanScramble`: Scramble and descramble binary input.
- `wlanBCCInterleave` and `wlanBCCDeinterleave`: Interleave and deinterleave binary convolutionally encoded input.
- `wlanBCCEncode` and `wlanBCCDecode`: Convolutionally encode and decode input.
- `wlanStreamParse` and `wlanStreamDeparse`: Stream-parse and deparse binary input.
- `wlanSegmentParseBits` and `wlanSegmentDeparseBits`: Segment-parse and deparse binary input.
- `wlanSegmentParseSymbols` and `wlanSegmentDeparseSymbols`: Segment-parse and deparse symbols.
- `wlanConstellationMap` and `wlanConstellationDemap`: Modulate and demodulate input.

You can explore these functions and customize their algorithms and functionality.

R2017a

Version: 1.3

New Features

Bug Fixes

Compatibility Considerations

R2017a

IEEE 802.11ad Support: Generate IEEE 802.11ad compliant waveforms

This release adds support for IEEE 802.11ad waveform generation. The 802.11ad specification defines the WLAN directional multi-gigabit (DMG) format. For details, see `wlanDMGConfig`.

IEEE TGah Indoor Channel Model: Simulate 802.11ah propagation conditions

In this release, WLAN System Toolbox provides the channel model described by the TGah task group. Use the `wlanTGahChannel` System object to model 802.11ah propagation conditions. For more information, see WLAN Channel Models and Propagation Channel.

OFDM Timing Synchronization: Detect and estimate symbol timing offsets in a received OFDM signal

A WLAN OFDM timing synchronization function,`wlanSymbolTimingEstimate`, returns the symbol timing estimate of a received OFDM waveform.

MATLAB Compiler Support: Compile WLAN models into standalone applications

WLAN System Toolbox now supports MATLAB Compiler™ enabling you to produce standalone executables. For details, see Code Generation and Deployment.

Scrambler Bit Mapping Change: Mapping of scrambler initialization seed is swapped

The scrambling operation now swaps the MSB-to-LSB mapping for the initialization seed for all supported WLAN packet formats. This change aligns the scrambler operation with the mapping indicated for 802.11ad. These functions perform the scrambling operation `wlanWaveformGenerator`, `wlanNonHTData`, `wlanHTData`, and `wlanVHTData`.

R2016b

Version: 1.2

New Features

IEEE 802.11ah Support: Generate IEEE 802.11ah compliant waveforms

This release adds support for IEEE 802.11ah waveform generation. For details, see `wlanS1GConfig` and 802.11ah Waveform Generation.

Multiuser-MIMO Receiver: Decode and analyze 802.11ac multiuser waveforms

This release adds support for IEEE 802.11ac MU-MIMO receiver. For details, see 802.11ac Multi-User MIMO Precoding.

LDPC Channel Coding: Analyze the performance of 802.11ac/n links using low-density parity-check (LDPC) channel coding techniques

This release adds LDPC support for IEEE 802.11ac and 802.11n. For details, see 802.11n Packet Error Rate Simulation for 2x2 TGn Channel.

Beacon Frame Generation Example: Create Beacon frames for receiver testing and over-the-air transmission

This release adds an example that creates and transmits a WLAN OFDM Beacon. You can transmit Beacon frames using an SDR platform, and then receive and view the Beacon using a standard Wi-Fi® device. For details, see 802.11 OFDM Beacon Receiver with USRP Hardware.

Productize functions

WLAN packet detection and format detection functions are productized in this release.

- Packet detection — Use the function `wlanPacketDetect` to return the offset from the start of the input waveform to the start of the detected preamble.
- Format detection — Use the function `wlanFormatDetect` to detect and return the packet format for the specified received signal.

R2016a

Version: 1.1

New Features

Bug Fixes

Support for 802.11p Standard: Simulate 802.11p systems to analyze Intelligent Transportation Systems (ITS) applications

The toolbox now supports IEEE 802.11p communications systems. Transmit and receive OFDM functions now include options for 10 MHz or 5 MHz channel bandwidth. For examples on how to use the 802.11p standard, see:

- 802.11p and 802.11a Packet Error Rate Simulations — Measures packet error rates in 802.11p and 802.11a links using an end-to-end simulation with a fading channel and AWGN.
- 802.11p Spectral Emission Mask Testing — Performs spectrum emission mask tests for an 802.11p transmitted waveform.

Support for 802.11j Standard: Simulate 802.11j systems to analyze Japanese WiFi market applications

Support is enabled for IEEE 802.11j™ communications systems. Transmit and receive OFDM functions now provide option for 10 MHz channel bandwidth.

Functionality Being Removed or Changed

Use of the `wlanGeneratorConfig` object is discouraged for parameterizing the `wlanWaveformGenerator` function. See `wlanWaveformGenerator` for the recommended parameter `Name,Value` pair syntax.

Functionality	Result	Use Instead	Compatibility Considerations
`wlanWaveformGenerator(bits,cfgFormat,cfgWaveGen)` where `cfgWaveGen` is a `wlanGeneratorConfig` object.	Still runs	`wlanWaveformGenerator(bits,cfgFormat,Name,Value,...,NameN,ValueN)`	Replace all instances of old function syntax with new function syntax.

R2015b+

Version: 1.0

New Features

IEEE 802.11ac and 802.11b/a/g/n standard-compliant physical layer models

WLAN System Toolbox provides standard-compliant functions for the design, simulation, and verification of IEEE 802.11 b/a/g/n/ac communications systems.

Very high throughput (VHT), high throughput (HT-mixed), and legacy (non-HT) waveform generation

WLAN System Toolbox provides 802.11 standard-compliant waveform generation for VHT, HT, and non-HT formats.

For more information, see Waveform Generation and Signal Transmission.

Channel coding, modulation (OFDM, DSSS, CCK), spatial stream mapping, and MIMO receivers

WLAN System Toolbox provides 802.11 standard-compliant functions for individual preamble field creation and decoding, OFDM demodulation and channel estimation, carrier frequency offset estimation, and data recovery. Single input single output (SISO) and multiple input multiple output (MIMO) antenna configurations are supported.

For more information, see Packet Recovery and Signal Reception.

Channel models, including TGac and TGn

WLAN System Toolbox provides channel models described by the TGac and TGn task groups. For more information, see WLAN Channel Models and Propagation Channel.

Measurements including channel power, spectrum mask, EVM, PER, and occupied bandwidth

WLAN System Toolbox examples show how to measure signal characteristics and system performance.

Waveform transmission and reception with radio devices and instruments

When coupled with a radio hardware support package, WLAN System Toolbox functions enable transmission and reception of packet contents with radio devices and instruments.

C code generation support

All WLAN System Toolbox functions support C code generation. Also, you can access the underlying code to customize the algorithms and functions.