Efficient Abstractions for Implementing TGn Channel and OFDM-MIMO Links in ns-3

Sian Jin
University of Washington
Seattle, WA, USA
sianjin@uw.edu

Sumit Roy
University of Washington
Seattle, WA, USA
sroy@uw.edu

Weihua Jiang
Xiamen University
Xiamen, China
whjiang@stu.xmu.edu.cn

Thomas R. Henderson
University of Washington
Seattle, WA, USA
tomh@tomh.org

ABSTRACT

Packet-level network simulators such as ns-3 require accurate physical (PHY) layer models for packet error rate (PER) for wideband transmission over fading wireless channels. To manage complexity and achieve practical runtimes, suitable link-to-system mappings can convert high fidelity PHY layer models for use by packet-level simulators. This work reports on two new contributions to the ns-3 Wi-Fi module, which presently only contains error models for Single Input Single Output (SISO), additive white Gaussian noise (AWGN) channels. To improve this, a complete implementation of a link-to-system mapping technique for IEEE 802.11 TGn fading channels is presented that involves a method for efficient generation of channel realizations within ns-3. The runtimes for the prior method suffers from scalability issues with increasing dimensionality of Multiple Input Multiple Output (MIMO) systems. We next propose a novel method to *directly* characterize the probability distribution of the"effective SNR" in link-to-system mapping. This approach is shown to require modest storage and not only reduces ns-3 runtime, it is also insensitive to growth of MIMO dimensionality. We describe the principles of this new method and provide details about its implementation, performance, and validation in ns-3.

CCS CONCEPTS

• **Computing methodologies** → **Modeling methodologies**; **Model verification and validation**; • **Networks** → **Wireless local area networks**.

KEYWORDS

Network Simulator 3 (ns-3), 802.11n, TGn, OFDM-MIMO, Link-to-system-mapping, Effective SNR, PER

WNS3 2020, June 17–18, 2020, Gaithersburg, MD, USA

ACM ISBN 978-1-4503-7537-5/20/06…$15.00
https://doi.org/10.1145/3389400.3389403

ACM Reference Format:
Sian Jin, Sumit Roy, Weihua Jiang, and Thomas R. Henderson. 2020. Efficient Abstractions for Implementing TGn Channel and OFDM-MIMO Links in ns-3. In *2020 Workshop on ns-3 (WNS3 2020), June 17–18, 2020, Gaithersburg, MD, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3389400.3389403

1 INTRODUCTION

IEEE TGn channel models [6] are widely adopted for simulating the performance of IEEE 802.11n protocols. These time-dispersive (multipath) channel models are typically modeled as "block fading" (i.e. the channel coefficients are assumed constant over a period called the "coherence interval") and described in the *time-domain* as a function of several key parameters: coherence time, total number of taps, delay and normalized power of each tap, Ricean K-factor of each tap and transmit/receive correlation matrix of each tap. The above time-domain TGn channels are usually further converted into frequency-domain TGn channels for most useful applications (e.g., received SNRs calculation). TGn channels in the frequency domain have frequency selectivity, which makes the received SNR on different subcarriers variable and fluctuating. While the link packet error ratio (PER) is largely dependent on the *worst* SNR over different subcarriers, the PER for the TGn channel is significantly degraded compared to a pure additive white Gaussian noise (AWGN) channel due to the frequency selective fading.

The mainline of ns-3 only supports AWGN PER models for the Wi-Fi module, although a previous effort [15] developed a link-to-system mapping framework for Orthogonal Frequency Division Multiplexing (OFDM) Single Input Single Output (SISO) modes for TGn channel models by using the commercial MATLAB WLAN System Toolbox. A limitation of the previous work was that there was no implementation of a channel generator within ns-3 to instantiate realizations of the frequency-selective channel. So running ns-3 simulation relied on pre-storing a large number of realizations of *frequency domain* channel matrices generated using the MATLAB WLAN toolbox, as detailed in [15]. In a simulation run, ns-3 would randomly draw a frequency domain channel matrix sample in each coherence interval. This requires a large number of realizations to be stored to estimate PERs from 10^{-2} or below. For example, for achieving PER accuracy at 10^{-2} within 95% confidence interval and 10% error bar, 40000 channel realizations should be stored [15]. Even for basic SISO systems, such an approach leads to large storage and

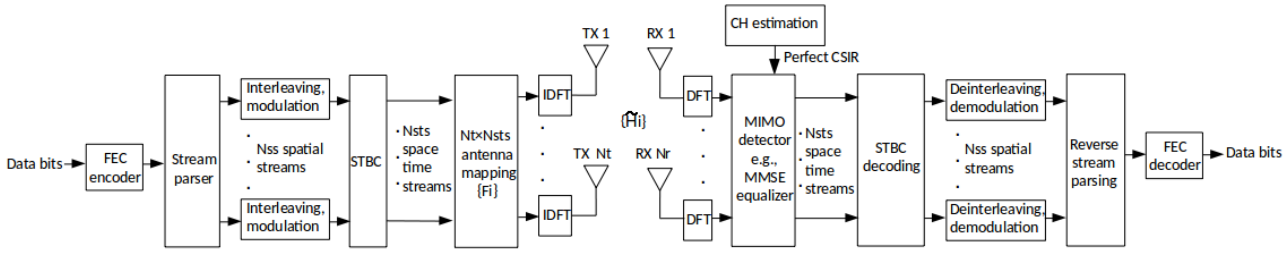

Figure 1: PHY Layer Block Diagram of the IEEE 802.11n OFDM-MIMO System (Modified from [16])

memory requirements, and as the number of sub-carriers (N_d) and the number of transmit/ receive antennas (N_t and N_r) increase, the memory cost of this method scales proportionately (see Table 1).

Table 1: Memory Cost of Storing 40000 Channel Model-C Realizations under a Single-run Using the Method in [15]

$N_t \times N_r$	Bandwidth	Storage Estimate
1×1	20MHz/40MHz	416.7 MB/847.7 MB
2×1	20MHz/40MHz	831.8 MB/1.7 GB
2×2	20MHz/40MHz	1.7 GB/3.4 GB
3×3	20MHz/40MHz	3.7 GB/7.6 GB
4×2	20MHz/40MHz	3.3 GB/6.8 GB

To avoid pre-storing large amount of channel instances in ns-3, this work provides a full implementation of generating TGn fading channel instances at run-time, leveraging Schumacher's channel generation code [20], thereby significantly reducing storage complexity over [15]. The PER for such links can be obtained using exponential effective SNR mapping (EESM) based link-to-system mapping method that converts received SNRs on different subcarriers and streams for OFDM-Multiple Input Multiple Output (MIMO) PHY into a *single* metric called "effective SNR", i.e., the SNR that would yield the same PER if the simulation was run for an AWGN channel. The effective SNR is mapped into PER using an equation based on the modulation type, or PER-SNR lookup table, as already implemented in ns-3. The EESM based link-to-system mapping previously adopted by [15] for OFDM-SISO system, is extended to IEEE 802.11n OFDM-MIMO system in Figure 1.

A detailed simulation model for OFDM-MIMO link performance evaluation is shown in Figure 2, which consists of two pieces: a pre-simulation part and a simulation part. A pre-simulation part either generates a large number of TGn specified frequency domain channel instances $\{\widetilde{H}_i\}_{i=1}^{N_d}$ for N_d subcarriers using the MATLAB TGn channel generator and stores generated channel instances in ns-3 as implemented in [15], or generates $\{\widetilde{H}_i\}_{i=1}^{N_d}$ directly within ns-3 using the TGn channel generator as implemented in this work. The generated channel instances are used to calculate EESM parameter β for link-to-system mapping. The pre-simulation part is done once, and the generated results (e.g., β) can be reused for future simulations. During simulation, ns-3 randomly draws/generates a frequency domain channel matrix sample $\{\widetilde{H}_i\}_{i=1}^{N_d}$ in each coherence interval. The channel matrix $\{\widetilde{H}_i\}_{i=1}^{N_d}$ and physical (PHY) layer processing jointly dictate the received SNR vector $\{\gamma_{i,j}\}_{i=1,j=1}^{N_d,N_{ss}}$ for N_d subcarriers and N_{ss} MIMO spatial streams. For a given received SNR vector $\{\gamma_{i,j}\}_{i=1,j=1}^{N_d,N_{ss}}$, EESM is adopted to map $\{\gamma_{i,j}\}_{i=1,j=1}^{N_d,N_{ss}}$ into an effective SNR scalar γ_{eff}. The PER for the TGn channel may now be computed using γ_{eff} and a PER-SNR lookup table for an equivalent AWGN channel.

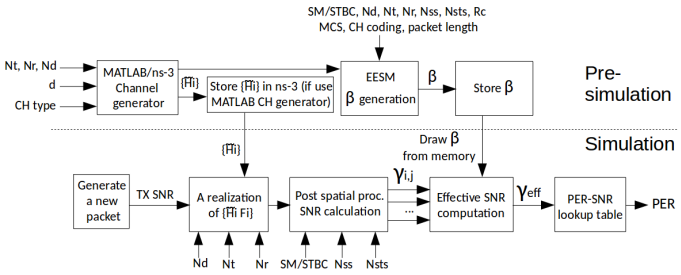

Figure 2: Simulation Model for Implementing Link-to-system Mapping in OFDM-MIMO System

Running the EESM based link-to-system mapping simulation as exemplified in Figure 2 achieves high accuracy, but requires large runtimes that scale with MIMO dimensionality and bandwidth. So, this work also provides a simpler abstraction to *directly* generate effective SNR from a suitable probability distribution function that is shown to be characterized by *few* parameters. The payoff will be shown in the results: the runtime of the simulation in Figure 2 scales as system dimensionality increases, while the runtime of the new approach is insensitive to the dimensionality change.

2 EESM-BASED LINK-TO-SYSTEM MAPPING UNDER TGN CHANNELS AND OFDM-MIMO SYSTEM

2.1 Overview of TGn Channel Models

The TGn channel models consist of six models (models A-F) corresponding to different indoor scenarios. Consider a transmiter-receiver (TX-RX) pair with N_t transmit antennas and N_r receive antennas, each TGn channel model can be specified as the following time domain MIMO channel matrix [6, 16]:

$$H(t,\tau) = \frac{1}{\sqrt{PL(d)}} \sum_{l=0}^{L-1} H_l(t)\delta(\tau - \tau_l),\ 0 \le t \le T, \qquad (1)$$

where $PL(d)$ is the large scale path loss (in linear scale), $\delta(.)$ is the Dirac delta function, L is the total number of taps, matrix $H_l(t)$ is a $N_r \times N_t$ MIMO channel coefficient matrix of l-th tap, τ_l is delay of l-th tap, and T is the coherence time.

2.1.1 Large Scale Path Loss Model. Large-scale path loss refers to the average loss in signal strength over distance [16]. In (1), large scale path loss $PL(d)$ includes free space path loss and a shadow fading loss due to large scale obstruction [18]. For TX-RX separation distance d measured in meters and central frequency f measured in hertz, large scale path loss follows the breakpoint model:

$$PL(d) = 20\log_{10} d + 20\log_{10} f - 147.55 + \begin{cases} X_\sigma, & d \le d_{BP} \\ 35\log_{10}(d/d_{BP}) + X_\sigma, & d > d_{BP} \end{cases}$$

where $PL(d)$ is in dB, d_{BP} is the break-point distance, X_σ is the gain (in dB) of random shadow fading and follows normal distribution. It follows that a TGn channel is in line-of-sight (LOS) condition if $d \le d_{BP}$, and it is in non-line-of-sight (NLOS) condition if $d > d_{BP}$. The channel conditions influence the small scale fading model discussed later.

2.1.2 Time Domain Small Scale Fading Model. The small scale fading properties of channel models are typically categorized in terms of their Doppler spread and their multipath delay spread [19]. From a Doppler spread perspective, all TGn channel models correspond to slow fading (coherence time much greater than symbol time). By [6], the coherence time of TGn channels is $T = \frac{3\ln(2)\lambda}{2\pi v_0}$, where λ is the wavelength, and v_0 is the relative speed of environmental scatterers (defined as 1.2 km/h in [6]). For center frequency f_c = 5GHz, the coherence time T = 66.2ms, which is much larger than IEEE 802.11 symbol duration of 3.6 or 4us. Using this property, the TGn small scale fading can be modeled as "block fading", which is static within a coherence interval. In simulators like ns-3, the channel matrix in (1) only need be updated once for each coherence interval.

From a multipath delay spread perspective, TGn channel models include both flat (non frequency selective) fading (model A) or frequency-selective fading (models B-F). The multipath delay is reflected by $\sum_{l=0}^{L-1} H_l(t)\delta(\tau - \tau_l)$ in (1). The channel coefficient matrix $H_l(t)$ is the sum of a LOS component, deterministic matrix $H_{l,LOS}(t)$ and a NLOS component, Rayleigh random matrix $H_{l,NLOS}$ [16]:

$$H_l(t) = H_{l,LOS}(t) + H_{l,NLOS}. \tag{2}$$

Under NLOS condition, the LOS component is set to be 0 (i.e., $H_{l,LOS}(t) = 0$). Under LOS condition, the matrix $H_{l,LOS}(t)$ is based on the uniform linear array (ULA) configuration, and is given by [21]

$$H_{l,LOS}(t) = \sqrt{P_l}\sqrt{\frac{K_l}{K_l+1}}\exp[2\pi j\frac{v_0}{\lambda}t\cos(\pi/4)]S, \tag{3}$$

where P_l is the normalized power at channel tap l, K_l is the Ricean K-factor of channel tap l, and S is the Rice steering matrix:

$$S = \left[1, e^{2\pi j\frac{d_{rx}}{\lambda}\sin(AoA_{LOS})}, \ldots, e^{2\pi j(N_r-1)\frac{d_{rx}}{\lambda}\sin(AoA_{LOS})}\right]^T \times \left[1, e^{2\pi j\frac{d_{tx}}{\lambda}\sin(AoD_{LOS})}, \ldots, e^{2\pi j(N_t-1)\frac{d_{tx}}{\lambda}\sin(AoD_{LOS})}\right] \tag{4}$$

where d_{tx} and d_{rx} are transmit and receive antenna spacing of ULA, AoA_{LOS} and AoD_{LOS} are the angle of arrival and angle of departure of the LOS component. For TGn channel models, $AoA_{LOS} = AoD_{LOS} = \pi/4$ [6]. The calculation of $\{P_l\}_{l=1}^{L}$ relates with the cluster modeling of TGn channel as well as the channel condition, and is not discussed further due to space limitations; readers are referred to [20].

The Rayleigh random NLOS matrix $H_{l,NLOS}$ is based on TGn antenna correlation model, and is given by [16]

$$H_{l,NLOS} = \sqrt{P_l}\sqrt{\frac{1}{K_l+1}}\text{vec}^{-1}\left(\sqrt{R_{l,tx}\otimes R_{l,rx}}H_{l,iid}\right), \tag{5}$$

where $R_{l,tx}$ is the $N_t \times N_t$ transmit correlation matrix of tap l given in [22], $R_{l,rx}$ is the $N_r \times N_r$ receive correlation matrix of tap l given in [22], $\otimes$ indicates the Kronecker product, $\text{vec}^{-1}()$ indicates inverse vectorization, and $H_{l,iid}$ is a $N_r N_t \times 1$ vector of independent, unit variance, zero-mean, complex Gaussian random variables of tap l. A good MATLAB implementation of $R_{l,tx}$ and $R_{l,rx}$ can be found in [20].

2.1.3 Frequency Domain MIMO Channel Matrix. Assuming channel is static within a coherence interval, the time domain $N_r \times N_t$ MIMO channel matrix in (1) can be easily converted into a $N_r \times N_t$ frequency-domain MIMO channel matrix using Fourier transform [17]:

$$\widetilde{H}_i(t) = \int_{\tau=0}^{\infty} H(t,\tau)e^{-j2\pi f_i\tau}d\tau = \frac{1}{\sqrt{PL(d)}}\sum_{l=0}^{L-1} H_l(t)e^{-j2\pi f_i\tau_l},\ 0 \le t \le T, i = 1, \ldots, N_d \tag{6}$$

where f_i is the i-th subcarrier frequency, and N_d is the number of subcarriers. For future received SNRs calculation, $\{\widetilde{H}_i(t)\}_{i=1}^{N_d}$ in (6) is more commonly used than $H(t,\tau)$ in (1). In the following sections, we focus on a single coherence time interval and use $\widetilde{H}_i$ instead of $\widetilde{H}_i(t)$.

2.2 Overview of EESM

Exponential effective SNR mapping (EESM) is a link-to-system mapping method that maps PHY/channel configurations and received SNR on different subcarriers onto an effective SNR, which could be associated to the PER by means of PER-SNR look-up tables obtained by off-line simulations. A typical EESM implementation diagram is shown in Figure 2.

In SISO case ($N_t = N_r = 1$) as in [15], the channel matrix $\widetilde{H}_i$ reduces to a scalar, and the received SNR on subcarrier i is given by

$$\gamma_i = \frac{P_t}{\sigma_i^2}|\widetilde{H}_i|^2,\ i = 1, \ldots, N_d \tag{7}$$

where P_t is transmission power, and σ_i^2 is additive noise power that is assumed to be the same over different subcarriers. EESM

converts the vector of SNRs in (7) into an effective SNR for ease of measuring link performance. The formula of EESM based effective SNR in the SISO case is given by

$$\gamma_{eff} = -\beta \ln\left(\frac{1}{N_d}\sum_{i=1}^{N_d}\exp\left(-\frac{\gamma_i}{\beta}\right)\right), \tag{8}$$

where β is the EESM parameter. The EESM parameter β is chosen so that the PER under the OFDM system and channel $\{\widetilde{H}_i\}_{i=1}^{N_d}$ with received SNRs $\{\gamma_i\}_{i=1}^{N_d}$ is equivalent to the PER under an AWGN channel with received SNR γ_{eff}. The EESM parameter β depends on PHY layer configurations (MCS, antenna configuration, bandwidth, channel coding, packet length), and can be obtained using the process reviewed in [15]. If γ_{eff} is obtained, we can obtain PER via PER-SNR lookup table for the AWGN channel at the considered MCS, channel coding scheme (BCC/LDPC) and packet length.

We extend the above EESM method to the OFDM-MIMO system in Figure 1 with perfect channel state information at the receiver (CSIR), but without channel state information at the transmitter (CSIT). At the transmitter, a forward error correction (FEC) encoded data stream is divided into N_{ss} spatial streams by a stream parser. The N_{ss} spatial streams represent N_{ss} independent information flows. The N_{ss} spatial streams are then coded into N_{sts} space-time streams using space-time-block-coding (STBC). For example, with Alamouti STBC, one single spatial stream is coded into two space-time streams [18]. The N_{sts} space-time streams are then mapped into N_t transmit antennas using a $N_t \times N_{sts}$ antenna mapping matrix F_i for subcarrier i. As CSIT is not available, the transmitter cannot do any beamforming/precoding, and F_i cannot be beamforming/precoding matrix. If $N_{sts} = N_t$, direct mapping is usually adopted and $F_i = I$. If $N_{sts} < N_t$, spatial expansion is usually adopted and F_i is provided by Section 20.3.11.11.2 of [1]. At the receiver, MIMO detection is used to map received signal from N_r receive antennas into N_{sts} space-time streams, which are further decoded by STBC decoder to recover N_{ss} spatial streams, where

$$1 \le N_{ss} \le N_{sts} \le \min\{N_t, N_r\}. \tag{9}$$

EESM based effective SNR for OFDM-MIMO system is a simple extension of OFDM-SISO relation (8) [10, 11]:

$$\gamma_{eff} = -\beta \ln\left(\frac{1}{N_d}\frac{1}{N_{ss}}\sum_{i=1}^{N_d}\sum_{j=1}^{N_{ss}}\exp\left(-\frac{\gamma_{i,j}}{\beta}\right)\right), \tag{10}$$

where $\gamma_{i,j}$ represents the post-processing SNR on subcarrier i on stream j. Here, $\gamma_{i,j}$ and β depend on the PHY and channel configurations.

If pure spatial multiplexing (SM) is adopted, $1 \le N_{ss} \le \min\{N_t, N_r\}$ independent streams are transmitted simultaneously in a symbol time and the number of space-time streams $N_{sts} = N_{ss}$. For minimum mean-squared error (MMSE) equalizer based MIMO decoding, $\gamma_{i,j}$ is given by [9]

$$\gamma_{i,j} = \frac{1}{[(\frac{P_t}{N_t\sigma_i^2}F_i^*\widetilde{H}_i^*\widetilde{H}_iF_i + I)^{-1}]_{j,j}} - 1, 1 \le i \le N_d, 1 \le j \le N_{ss}. \tag{11}$$

If pure space time block coding (STBC) is adopted with spatial rate R_c, R_cT_c symbols are transmitted over T_c symbol time, the number of spatial streams $N_{ss} = 1$, and $1 \le N_{sts} \le \min\{N_t, N_r\}$. If maximum likelihood (ML) decoder is used for STBC decoding, the post-processing SNR $\gamma_{i,j}$ is given by [13]

$$\gamma_{i,j} = \frac{P_t}{N_t\sigma_i^2R_c}||\widetilde{H}_iF_i||_F^2, 1 \le i \le N_d, j = 1. \tag{12}$$

where $||\widetilde{H}_iF_i||_F$ denotes the Frobenius norm of $\widetilde{H}_iF_i$.

2.3 Implementing EESM-based Link-to-system Mapping in ns-3

The block diagram for implementing EESM in OFDM-MIMO system is shown in Figure 3. The TGn channel generator in Figure 3 generates TGn channel using a standard time domain TGn channel model described in (1) - (5). The path loss $PL(d)$ is implemented in a new class *TgnPropagationLossModel*. Other steps in (1) - (5) are implemented in a new class *TgnChannelModel*. Then, TGn channel generator converts the standard time domain TGn channel model into a frequency domain channel $\{\widetilde{H}_i\}_{i=1}^{N_d}$ using (6). This is implemented in another new class *TgnPropagationFadingModel*. The simulation parts of Figure 3 uses (10) - (12) to calculate post-processing SNRs and then obtain effective SNR. This is presently implemented outside of the wifi module but is planned to be added to the class *InterferenceHelper*. All ns-3 codes implemented for this paper are available online [12].

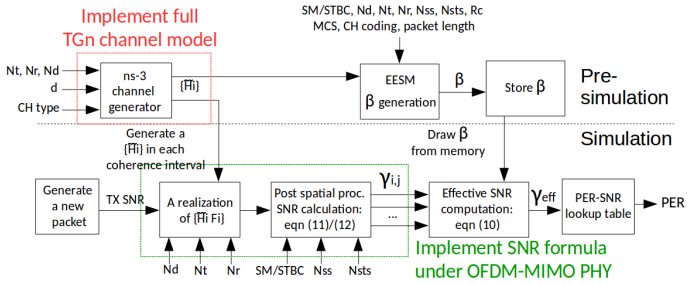

Figure 3: Block Diagram for Implementing EESM in OFDM-MIMO System

The class *TgnPropagationLossModel* contains *GetPathLossDb*, which calculates free space path loss in dB, and *GetShadowLossDb* which calculates shadow fading loss in dB. The path loss $PL(d)$ in dB is the sum of the returns of *GetPathLossDb* and *GetShadowLossDb*.

The class *TgnChannelModel* contains the following:

- Structure *ParamsTgn* stores parameters of the TGn channels. The parameters include number of taps, tap delay, normalized tap power and Ricean K-factor. The parameters also include N_t, N_r and the correlation matrix $\sqrt{R_{l,tx} \otimes R_{l,rx}}$. Correlation matrices are generated using Schumacher's MATLAB code [20], and are saved as static matrices.
- Structure *TgnChannelMatrix* stores a temporary realization of the standard time domain TGn channel and parameters including number of taps, tap delay, N_t and N_r.
- Method *SetTgnTable* sets *ParamsTgn* according to the type of TGn channel, N_t and N_r, inputted by users. N_t and N_r are obtained from ns-3-dev class *AntennaArrayBasicModel* [12].

- Method *GetChannel* generates the standard time domain TGn channel matrix using (1) - (5). The input of *GetChannel* is *ParamsTgn*, and the output of *GetChannel* is *TgnChannelMatrix*.

The class *TgnPropagationFadingModel* contains the following:

- Method *CalFreqChannel* convert the standard time domain TGn channel matrix generated from *TgnChannelModel* into a frequency domain TGn channel matrix $\{\widetilde{H}_i\}_{i=1}^{N_d}$ using (6).
- Method *GetFreqChannel* returns previous $\{\widetilde{H}_i\}_{i=1}^{N_d}$ when the coherence time is not over. It calls *CalFreqChannel* to update $\{\widetilde{H}_i\}_{i=1}^{N_d}$ and then returns a new $\{\widetilde{H}_i\}_{i=1}^{N_d}$ when the coherence time is over.

The class *InterferenceHelper* can be extended with a method *GetEffectiveSnr*. When pure SM is adopted, this method calculates post-processing SNR $\{\gamma_{i,j}\}_{i=1,j=1}^{N_d,N_{ss}}$ using (11); when pure STBC is adopted, this method calculates post-processing SNR $\{\gamma_{i,j}\}_{i=1,j=1}^{N_d,N_{ss}}$ using (12). Then, γ_{eff} is calculated based on $\{\gamma_{i,j}\}_{i=1,j=1}^{N_d,N_{ss}}$ and (10).

2.4 Simulation Runtime Evaluation

We compare the runtimes of the EESM based link-to-system mapping as exemplified in Figure 3 on ns-3 and MATLAB WLAN Toolbox full link simulations under channel model-C with LOS condition. 40000 packets are generated in both simulations. The runtimes start from the generation of the TGn channel till the calculation of γ_{eff}. All simulations are conducted on a Intel(R) Xeon(R) E5-2630 CPU at 2.40GHz. To speed up ns-3 simulation, ns-3 is configured to build optimized libraries. From Table 2 that reports simulation runtimes, we have the following observations:

- The EESM based simulation runs faster than MATLAB full link simulations, due to efficient link-to-system mapping.
- For bandwidth scaling while keeping (N_t, N_r) constant, the runtime of the EESM error model increases (almost) linearly with bandwidth. Since the number of subcarriers N_d increases proportionally with bandwidth, the number of additional loops for calculating γ_{eff} in (10) contributes to this increase.
- For fixed bandwidth, the runtimes of the link-to-system mapping in Figure 3 for 1×1 case with 1 stream, 2×2 case with 2 streams, 3×3 case with 3 streams etc. scale due to the increase in N_t, N_r, and the resulting computation in (10), (11).

Table 2: Single-run Runtimes Comparison between the Full Link Simulation on MATLAB WLAN Toolbox and the EESM-based Link-to-system Mapping in ns-3

$N_t \times N_r (= N_{ss})$	Bandwidth	MATLAB Full-link	ns-3 EESM
1×1	20MHz	28 min	13.34 s
1×1	40MHz	25 min	24.97 s
2×2	20MHz	37 min	25.45 s
2×2	40MHz	39 min	48.82 s
3×3	20MHz	51 min	41.37 s
3×3	40MHz	60 min	77.63 s

In summary, while the runtimes for a single link using the EESM based link-to-system mapping in Figure 3 are much smaller than MATLAB full-link simulation, it still scales with system parameters N_d, N_t, N_r. Newer Wi-Fi standards scale MIMO dimensions N_t, N_r to exceed 8, 8 on 80 MHz/160 MHz channels, implying that a more efficient technique will be needed for effective ns-3 runtimes.

3 NEW EFFICIENT SIMULATION METHOD VIA STATISTICAL CHARACTERIZATION OF γ_{eff}

3.1 Modeling Distribution of γ_{eff}

An alternative to the simulation method in Figure 3 is to statistically characterize the distribution of γ_{eff} using a few model dependent parameters and use it *directly* in link-to-system mapping. The proposed new statistical method results in an efficient simulation technique, relative to current practice.

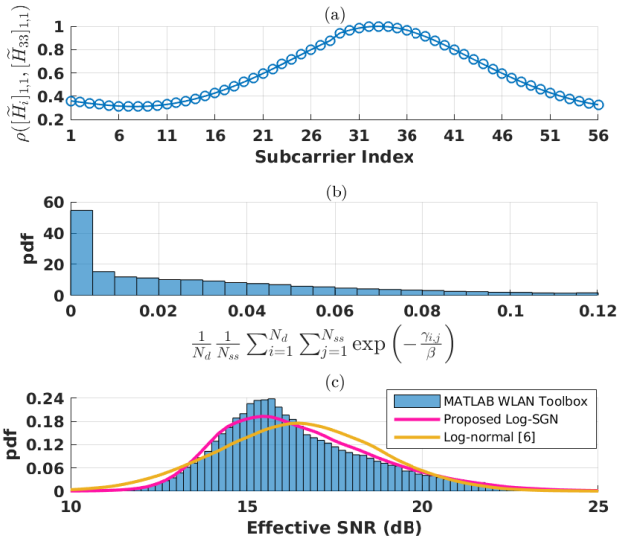

Figure 4: Simulating (a) the Correlation Coefficient between Different Subcarrier Gains (b) the pdf of $\frac{1}{N_d}\frac{1}{N_{ss}}\sum_{i=1}^{N_d}\sum_{j=1}^{N_{ss}}\exp\left(-\frac{\gamma_{i,j}}{\beta}\right)$ (c) the pdf of γ_{eff}

We first focus on finding the approximate distribution of γ_{eff} under TGn channels, based on some evidences from our simulations in Figure 4 and related works. Our simulation is run once over 40000 different channel instances, under channel model-C with LOS condition, 2×2 SM with 2 streams, 20MHz bandwidth, BCC channel coding, MCS = 12, transmit SNR $\frac{P_t}{\sigma_i^2} = 30.4$dB without applying path loss. First, under TGn MIMO channel, even when $|i-j|$ is moderately large, the correlation coefficient between $[\widetilde{H}_i]_{n_r,n_t}$ and $[\widetilde{H}_j]_{n_r,n_t}$ (i.e., $\rho([\widetilde{H}_i]_{n_r,n_t}, [\widetilde{H}_j]_{n_r,n_t})$) is still large. For example, in Figure 4 (a), $\rho([\widetilde{H}_i]_{1,1}, [\widetilde{H}_j]_{1,1}) \geq 0.6$ when $|i-j| \leq 12$. Since $\widetilde{H}_i, i = 1, \ldots, N_d$ do not satisfy the weakly dependent condition in [3] and further N_d is not sufficiently large, central limit theorem (CLT) is not an accurate representation for $\frac{1}{N_d}\frac{1}{N_{ss}}\sum_{i=1}^{N_d}\sum_{j=1}^{N_{ss}}\exp\left(-\frac{\gamma_{i,j}}{\beta}\right)$

in (10), whose pdfs by simulation in Figure 4 (b) under TGn MIMO channels, are highly skewed (closer to Chi-square distribution under some special cases). Thus, the normal or Beta approximation for $\frac{1}{N_d}\frac{1}{N_{ss}}\sum_{i=1}^{N_d}\sum_{j=1}^{N_{ss}}\exp\left(-\frac{\gamma_{i,j}}{\beta}\right)$ based on CLT [7, 14, 23] no longer holds under TGn channels.

However, [5] shows empirically that γ_{eff} can be approximated by log-normal random variable (i.e., $\ln(\gamma_{eff})$ is normally distributed), and this approximation is superior to modeling $\frac{\sum_{i=1}^{N_d}\sum_{j=1}^{N_{ss}}\exp\left(-\frac{\gamma_{i,j}}{\beta}\right)}{N_d N_{ss}}$ as a Chi-square/normal random variable. In Figure 4 (c), we fit the pdf of γ_{eff} to log-normal pdf using method of moments, and find that modeling $\ln(\gamma_{eff})$ as normal random variable is still not desirable. The main reason is that the normal random variable has only two parameters that can control mean and variance, but lacks flexibility in simultaneously controlling mean, variance, and shape (e.g., skewness and kurtosis). To achieve better control of shape, we model $\ln(\gamma_{eff})$ as a skew-generalized normal (SGN) random variable [2] (referred to as log-SGN approximation of γ_{eff}), i.e.,

$$X \triangleq \ln(\gamma_{eff}) \sim \text{SGN}(\hat{\mu}, \hat{\sigma}, \hat{\lambda}_1, \hat{\lambda}_2), \tag{13}$$

with pdf

$$f_X(x; \hat{\mu}, \hat{\sigma}, \hat{\lambda}_1, \hat{\lambda}_2) = \frac{2}{\hat{\sigma}}\phi\left(\frac{x-\hat{\mu}}{\hat{\sigma}}\right)\Phi\left(\frac{\hat{\lambda}_1(x-\hat{\mu})}{\sqrt{\hat{\sigma}^2+\hat{\lambda}_2(x-\hat{\mu})^2}}\right),\ x \in \mathbf{R}, \tag{14}$$

where $\hat{\mu} \in \mathbf{R}$ is the location parameter, $\hat{\sigma} > 0$ is the scale parameter, $\hat{\lambda}_1 \in \mathbf{R}$ and $\hat{\lambda}_2 \geq 0$ are shape parameters, $\phi(x)$ is the standard normal pdf and $\Phi(x)$ is the standard normal cdf. Modeling X (i.e., $\ln(\gamma_{eff})$) via SGN random variable has two advantages. First, it simultaneously controls mean, variance and shape (e.g., skewness and kurtosis), and is desirable for fitting γ_{eff} that is skewed resulting from TGn MIMO channels (see Figure 4 (c)). Second, SGN variables are easy to generate, as will be shown in Algorithm 2.

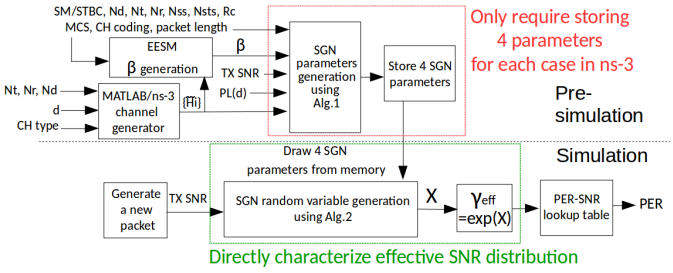

Figure 5: Proposed Statistical Method for Implementing EESM in OFDM-MIMO System

The block diagram of our new method for implementing EESM in OFDM-MIMO system is shown in Figure 5. The workflow consists of two stages: a pre-simulation stage and a simulation stage. The pre-simulation stage generates a large number of channel instances from a MATLAB channel generator (or generates channel instances in ns-3 as implemented in this work) and uses these channel instances to calculate the EESM parameter β and the SGN parameters $\hat{\mu}, \hat{\sigma}, \hat{\lambda}_1, \hat{\lambda}_2$. Only $\hat{\mu}, \hat{\sigma}, \hat{\lambda}_1, \hat{\lambda}_2$ (and no channel realizations - unlike [15]) are stored in the pre-simulation step. The steps for obtaining $\hat{\mu}, \hat{\sigma}, \hat{\lambda}_1, \hat{\lambda}_2$ are shown in Algorithm 1. In the simulation stage, a realization of SGN distributed X can be generated using Algorithm 2 with $\hat{\mu}, \hat{\sigma}, \hat{\lambda}_1, \hat{\lambda}_2$ as inputs in each coherence interval. The realization of X is then used to calculate $\gamma_{eff} = \exp(X)$. In summary, reducing a complicated PHY/channel setup to four parameters is the fundamental innovation, that makes the runtime and storage complexity largely insensitive to the dimension of the PHY layer parameters (e.g., N_d, N_t, N_r, N_{ss} and N_{sts}); changing these parameters only changes the values of $\hat{\mu}, \hat{\sigma}, \hat{\lambda}_1, \hat{\lambda}_2$, which almost does not impact the runtime and storage complexity.

Algorithm 1: Obtain Parameters of SGN Distributed X

Input: MIMO setup (SM/STBC, $N_t, N_r, N_{ss}, N_{sts}, R_c$), bandwidth, MCS, channel coding type (BCC/LDPC), packet length (short/long), $\frac{P_t}{\sigma_i^2}\frac{1}{PL(d)}$ and TGn channel type (A-F).
Output: $\hat{\mu}, \hat{\sigma}, \hat{\lambda}_1, \hat{\lambda}_2$.

1: Obtain n realizations of frequency-domain MIMO channel matrices $\{\widetilde{H}_i\}_{i=1}^{N_d}$ using MATLAB WLAN toolbox or ns-3 channel generator (implemented in Section 2.3), where n is large.
2: **if** pure SM is chosen **then**
3: Generate n realizations of X (i.e., $\{x_k\}_{k=1}^n$) using $\{\widetilde{H}_i\}_{i=1}^{N_d}$, (10), (11) and (13).
4: **end if**
5: **if** pure STBC is chosen **then**
6: Generate n realizations of X (i.e., $\{x_k\}_{k=1}^n$) using $\{\widetilde{H}_i\}_{i=1}^{N_d}$, (10), (12) and (13).
7: **end if**
8: Use MATLAB *fmincon* function to numerically maximize the log-likelihood function of the SGN random variable [2]:

$$-\frac{n}{2}\log\frac{\pi\hat{\sigma}^2}{2} - \frac{1}{2\hat{\sigma}^2}\sum_{k=1}^{n}(x_k-\hat{\mu})^2 + \sum_{k=1}^{n}\log\Phi\left(\frac{\hat{\lambda}_1(x_k-\hat{\mu})}{\sqrt{\hat{\sigma}^2+\hat{\lambda}_2(x_k-\hat{\mu})^2}}\right),$$

under the constraints $\hat{\sigma} > 0$ and $\hat{\lambda}_2 \geq 0$, and output the optimized solution.

Algorithm 2: Generate A Realization of SGN Distributed X [2, 8]

Input: $\hat{\mu}, \hat{\sigma}, \hat{\lambda}_1, \hat{\lambda}_2$ obtained from Algorithm 1.
Output: A realization of X.

1: Generate a Gaussian distributed random variable $\alpha = \sqrt{\hat{\lambda}_2}Z + \hat{\lambda}_1$, where $Z \sim \mathcal{N}(0,1)$.
2: Based on the realization of α, generate two i.i.d. Gaussian random variables $U1, U2 \sim \mathcal{N}\left(\sqrt{\frac{1+\alpha^2}{2}}\hat{\mu}, \hat{\sigma}^2\right)$.
3: Based on the realization of $U1$ and $U2$, let $U = \max(U1, U2)$ and $V = \min(U1, U2)$. Then, $X = \frac{1+\alpha}{\sqrt{2(1+\alpha^2)}}U + \frac{1-\alpha}{\sqrt{2(1+\alpha^2)}}V$.

3.2 Implementing Log-SGN Based Statistical Model in ns-3

A new ns-3 class *TgnErrorModelParameter* is used to store the SGN parameters $\hat{\mu}, \hat{\sigma}, \hat{\lambda}_1, \hat{\lambda}_2$ generated using Algorithm 1. Once these parameters are generated and stored, it can be reused by other users. For any given channel/PHY configuration (MCS, N_t, N_r, N_d), the four SGN parameters are sensitive to the change of received power (due to varying transmit power or path loss $PL(d)$ - see Figure 6), and hence requires storing the SGN parameter set under different received power. From Section 5 in [18] (Figures 5.8, 5.13 - 5.15) and our MATLAB simulations in Section 4 (Figure 8), we can see that varying transmit/receive SNR over 25dB range suffices for obtaining all reasonable PERs down to 10^{-2} for IEEE 802.11n system. Under channel model-C with LOS condition without path loss, 2×2 SM with 2 streams, 40MHz bandwidth, BCC channel coding and MCS = 12, our simulations in Figure 6 shows that 0.25dB changes to the transmit/received SNR result in discernible but not too significant change to effective SNR distribution. Thus it is enough to store $25/0.25 = 100$ sets of SGN parameters for every 0.25dB SNR increment.

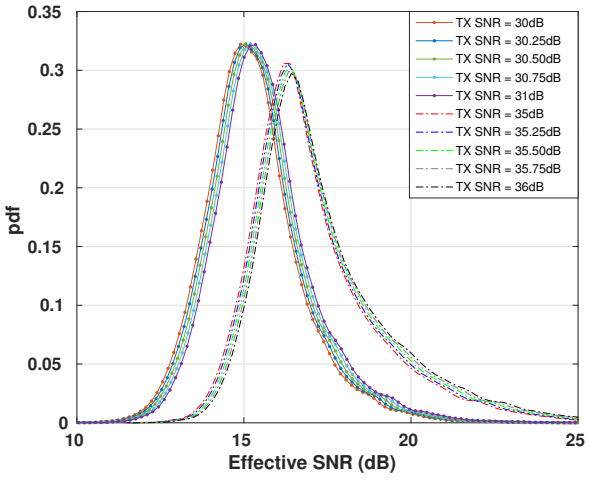

Figure 6: The pdf of Effective SNR under Different Transmit SNR

The new method *GetEffectiveSnr* in class *InterferenceHelper* can be extended to support the log-SGN based statistical model, which uses Algorithm 2 to generate SGN distributed X, and outputs the effective SNR γ_{eff} based on X and (13). The effective SNR γ_{eff} is passed to a class *EesmErrorModel*, which includes a AWGN SNR-PER lookup table that maps the γ_{eff}, MCS, channel coding scheme (BCC/LDPC) and packet length into the PER.

4 VERIFICATION AND COMPARISON OF ERROR MODELS IN NS-3

We first verify our proposed log-SGN based model and model in Figure 3 by comparing the effective SNR pdf obtained from ns-3 simulations with the effective SNR pdf obtained from MATLAB WLAN Toolbox. Figure 7 shows the comparison with the same setup as in Figure 4. From Figure 7, we can see that the model in Figure 3 accurately generates the effective SNR pdf. The effective SNR distribution generated by log-SGN based model approximates the effective SNR distribution generated by MATLAB WLAN Toolbox. Admittedly, the range of kurtosis of SGN random variable is limited, and the center peak cannot be fitted well. There exists better distributions (e.g., extended SGN [4] with five parameters) that have larger range of kurtosis and can fit the central peak better. However, generating such random variables is hard when compared with generating SGN random variables.

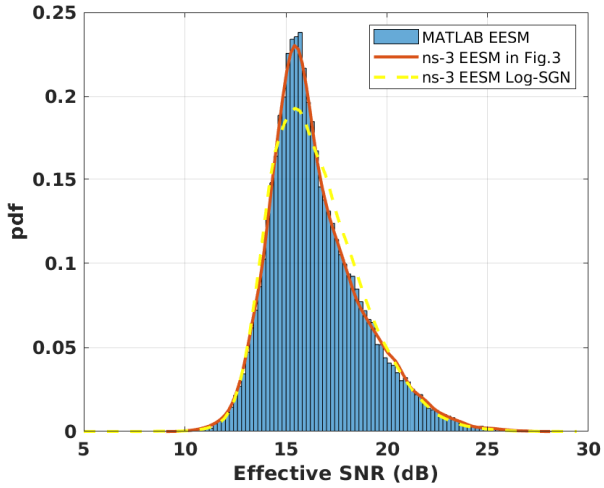

Figure 7: Comparison between the pdf Obtained from MATLAB WLAN Toolbox, the pdf Generated by the Model in Figure 3 and the Proposed Log-SGN Based Model

We next plot in Figure 8 the PER versus transmit SNR curve produced by link simulation on MATLAB WLAN Toolbox, EESM link-to-system mapping (using the model in Figure 3), and EESM link-to-system mapping using log-SGN based model under channel model-C with LOS condition, SM with full streams, 20MHz bandwidth, BCC channel coding, packet length = 1000 Byte and 16-QAM modulation with 3/4 channel coding rate. Path loss is not applied. The simulation is run once over 40000 different channel instances. From Figure 8, we can see that the link-to-system mappings from model in Figure 3 and log-SGN based model match well with MATLAB link simulation. Furthermore, the accuracy loss caused by log-SGN approximation against the implementation in Figure 3 is much smaller than the accuracy loss caused by EESM link-to-system mapping.

We then compare the storage complexity of the method in [15] and the proposed log-SGN method. For each channel type (A-F), channel bandwidth (20MHz/40MHz), N_r ($1 \sim 4$) and N_t ($1 \sim 4$), the proposed log-SGN based method requires storing 4 SGN parameters (32 Bytes) $\times$100 different received SNRs $\times$8 MCSs $\times$2 channel coding types (BCC/LDPC) $\times$2 packet lengths (short/long) $\times 4$ N_{ss} $\times$2 STBC rates ($R_c = 1$ for $N_{sts} = 2$ and $R_c = 1/2$ for $N_{sts} = 3, 4$ if STBC is applied; $R_c = 1$ if STBC is not applied) = 819.2 KB. This is much

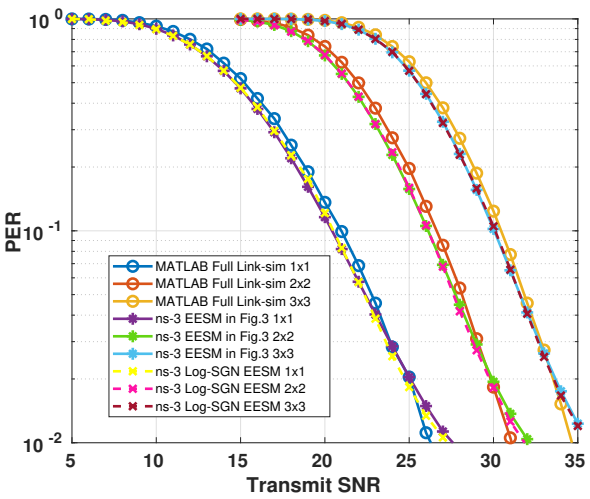

Figure 8: The PER versus Transmit SNR

smaller than the the method in [15], whose storage complexity is shown in Table 1. The memory cost of the proposed log-SGN method also does not scale with the increase of system parameters (N_t, N_r and bandwidth), as all channel matrices and PHY layer characteristics are finally mapped into four SGN parameters.

We finally compare the runtime of the log-SGN based model with the model in Figure 3 under the same setup as in Section 2.4 (running a simulation of 40000 packets once, same runtime calculation procedure). For any given N_d, N_t, N_r, the runtime of the log-SGN based model is always around 2.1 s, which is much smaller than the runtime of the model in Figure 3. The runtimes of the log-SGN based model do not scale with the system parameters (N_t, N_r and bandwidth), as all channel generation and PHY layer processing are mapped into the generation of X, whose runtime almost does not change with the change of the four SGN parameters.

5 CONCLUSION & FUTURE WORK

In this work, a full ns-3 implementation of TGn channels and a full link simulation abstraction based on EESM link-to-system mapping was described. It upgrades current ns-3 802.11 SISO system model to include 802.11n OFDM-MIMO use cases. Extending the prior SISO system to MIMO system preserves accuracy at the cost of significant increase in computation time, due to generating the TGn channel and OFDM-MIMO PHY layer processing. To manage this increase in simulation runtime, we next developed a new log-SGN based statistical model that *directly* characterizes the output effective SNR. This model requires only a few parameters to characterize the full link performance, is efficient in both memory and computation, and achieves good approximation to the results from the approach in Figure 3.

For future work, IEEE 802.11ac/ax PHY layer and TGac/TGax channels will be implemented using the proposed statistical methodology. Our preliminary investigations suggest that the proposed new method is even more suited for 11ac/ax system. TGac/TGax time-domain channels have larger number of taps L and 11ax design has smaller subcarrier spacing, which increases the computational burden for the method in Figure 3. In contrast, the SGN approximation of $\ln(\gamma_{eff})$ keeps the runtime low, as it is insensitive to the change of PHY parameters.

ACKNOWLEDGMENTS

This research was funded in part by NSF under ICE-T 1836725. The authors would also like to thank Laurent Schumacher for giving us permission to reuse TGn MATLAB code [20].

REFERENCES

[1] 2012. IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Redline. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007) - Redline* (March 2012), 1–5229.

[2] R. B. Arellano-Valle, H. W. Gómez, and F. A. Quintana. 2004. A New Class of Skew-normal Distributions. *Communications in statistics-Theory and Methods* 33, 7 (2004), 1465–1480.

[3] J-M. Bardet, P. Doukhan, G. Lang, and N. Ragache. 2008. Dependent Lindeberg Central Limit Theorem and Some Applications. *ESAIM: Probability and Statistics* 12 (2008), 154–172.

[4] K. Choudhury and M. A. Matin. 2011. Extended Skew Generalized Normal Distribution. *Metron* 69, 3 (2011), 265–278.

[5] S. N. Donthi and N. B. Mehta. 2011. An Accurate Model for EESM and Its Application to Analysis of CQI Feedback Schemes and Scheduling in LTE. *IEEE Transactions on Wireless Communications* 10, 10 (October 2011), 3436–3448.

[6] V. Erceg. 2004. IEEE 802.11 Wireless LANs: TGn Channel Models. (2004).

[7] J. Francis and N. B. Mehta. 2014. EESM-based Link Adaptation in Point-to-point and Multi-cell OFDM Systems: Modeling and Analysis. *IEEE Transactions on Wireless Communications* 13, 1 (January 2014), 407–417.

[8] D. Ghorbanzadeh, P. Durand, and L. Jaupi. 2017. Generating the Skew Normal Random Variable. In *Proceedings of the World Congress on Engineering*. London, UK.

[9] R. W. Heath Jr. and A. Lozano. 2018. *Foundations of MIMO Communication*. Cambridge University Press.

[10] R. P. F. Hoefel and O. Bejarano. 2016. On Application of PHY Layer Abstraction Techniques for System Level Simulation and Adaptive Modulation in IEEE 802.11 ac/ax Systems. *Journal of Communication and Information Systems* 31, 1 (2016).

[11] T. L. Jensen, S. Kant, J. Wehinger, and B. H. Fleury. 2010. Fast Link Adaptation for MIMO OFDM. *IEEE Transactions on Vehicular Technology* 59, 8 (Oct 2010), 3766–3778.

[12] S. Jin. 2020. Link-to-system-mapping-wns3-2020-code. *download information: https://depts.washington.edu/funlab/resources/* (2020).

[13] A. Maaref and S. Aissa. 2005. Capacity of Space-time Block Codes in MIMO Rayleigh Fading Channels with Adaptive Transmission and Estimation Errors. *IEEE Transactions on Wireless Communications* 4, 5 (Sep. 2005), 2568–2578.

[14] A. Oborina, V. Koivunen, and T. Henttonen. 2010. Effective SINR distribution in MIMO OFDM systems. In *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*. Pacific Grove, CA, USA.

[15] R. Patidar, S. Roy, T. R. Henderson, and A. Chandramohan. 2017. Link-to-system Mapping for ns-3 Wi-Fi OFDM Error Models. In *Proceedings of the Workshop on ns-3 (WNS3 '17)*. ACM, New York, NY, USA, 31–38.

[16] T. Paul and T. Ogunfrunmiri. 2008. Wireless LAN Comes of Age: Understanding the IEEE 802.11n Amendment. *IEEE Circuits and Systems Magazine* 8, 1 (First 2008), 28–54.

[17] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bolcskei. 2004. An Overview of MIMO Communications - A Key to Gigabit Wireless. *Proc. IEEE* 92, 2 (Feb 2004), 198–218.

[18] E. Perahia and R. Stacey. 2013. *Next Generation Wireless LANs: 802.11n and 802.11ac* (2nd ed.). Cambridge University Press, New York, NY, USA.

[19] T. S. Rappaport. 2002. *Wireless Communications: Principles and Practice* (2nd ed.). Prentice Hall.

[20] L. Schumacher. 2003. WLAN MIMO Channel MATLAB Program. *download information: http://www. info. fundp. ac. be/ lsc/Research/IEEE* (2003).

[21] L. Schumacher and B. Dijkstra. 2004. Description of a MATLAB Implementation of the Indoor MIMO WLAN Channel Model Proposed by the IEEE 802.11 TGn Channel Model Special Committee. *Implementation note version* 5 (2004).

[22] L. Schumacher, K. I. Pedersen, and P. E. Mogensen. 2002. From Antenna Spacings to Theoretical Capacities - Guidelines for Simulating MIMO Systems. In *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Vol. 2. Pavilhao Altantico, Portugal.

[23] H. Song, R. Kwan, and J. Zhang. 2011. Approximations of EESM Effective SNR Distribution. *IEEE Transactions on Communications* 59, 2 (February 2011), 603–612.