IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11ax Evaluation Methodology | | | | |
| Date: 2015-07-16 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Ron Porat | Broadcom | 16340 West Bernardo Dr., San Diego, CA 92127 | 858-521-5409 | [rporat@broadcom.com](mailto:rporat@broadcom.com) |
| Matt Fischer | Broadcom |  |  |  |
| Simone Merlin | Qualcomm |  |  |  |
| Sameer Vermani | Qualcomm |  |  |  |
| Edward Au | Huawei |  |  |  |
| David Yangxun | Huawei |  |  |  |
| Jiayin Zhang | Huawei |  |  |  |
| Jun Luo | Huawei |  |  |  |
| Jiyong Pang | Huawei |  |  |  |
| Robert Stacy | Intel |  |  |  |
| Shahrnaz Azizi | Intel |  |  |  |
| Chittabrata Ghosh | Intel |  |  |  |
| Wookbong Lee | LGE |  |  |  |
| HanGyu Cho | LGE |  |  |  |
| Jianhan Liu | Mediatek |  |  |  |
| James Yee | Mediatek |  |  |  |
| Laurent Cariou | Orange |  |  |  |
| Thomas Derham | Orange |  |  |  |
| Yasuhiko Inoue | NTT |  |  |  |
| Yusuke Asai | NTT |  |  |  |
| Yasushi Takatori | NTT |  |  |  |
| Akira Kishida | NTT |  |  |  |
| Koichi Ishihara | NTT |  |  |  |
| Akira Yamada | NTT DoCoMo |  |  |  |
| Shoko Shinohara | NTT |  |  |  |
| Masashi Iwabuchi | NTT |  |  |  |
| Sayantan Choudhury | Nokia |  |  |  |
| Esa Tuomaala | Nokia |  |  |  |
| Klaus Doppler | Nokia |  |  |  |
| Jarkko Kneckt | Nokia |  |  |  |
| Minho Cheong | ETRI |  |  |  |
| Jae Seung | ETRI |  |  |  |
| Leif Wilhelmsson | Ericsson |  |  |  |
| Filip Mestanov | Ericsson |  |  |  |
| Yakun Sun | Marvell |  |  |  |
| Jinjing Jiang | Marvell |  |  |  |
| Yan Zhang | Marvell |  |  |  |
| Kaushik Josiam | Samsung |  |  |  |
| Yonggang Fang | ZTE |  |  |  |
| Bo Sun | ZTE |  |  |  |
| Kaiying Lv | ZTE |  |  |  |
| Zhendong Lou | CATR |  |  |  |
| Meng Yang | CATR |  |  |  |
| Reza Hedayat | Newracom |  |  |  |
| Daewon Lee | Newracom |  |  |  |
| Brian Hart | Cisco |  |  |  |
| David Halls | Toshiba |  |  |  |
| Eric Wong | Apple |  |  |  |
| Joonsuk Kim | Apple |  |  |  |
| Guoqing Li | Apple |  |  |  |
| Xiaofei Wang | InterDigital |  |  |  |

# Abstract

This document describes the simulation methodology, evaluation metrics and traffic models for assessing 80.11ax proposals’ performance.

# Table of Contents

[Abstract 2](#_Toc409064652)

[Table of Contents 3](#_Toc409064653)

[Revisions 3](#_Toc409064654)

[Notes on this version 3](#_Toc409064655)

[Simulation Methodologies - General Concept 4](#_Toc409064656)

[System Simulation – High Level Description 5](#_Toc409064657)

[PER Simulation Description 6](#_Toc409064658)

[PHY System Simulation Detailed Description 6](#_Toc409064659)

[Integrated System Simulation Detailed Description 7](#_Toc409064660)

[MAC System Simulation Description 12](#_Toc409064661)

[Simulation Methodology Choice 12](#_Toc409064662)

[System Simulation Calibration 13](#_Toc409064663)

[Traffic Models 23](#_Toc409064664)

[Metrics 24](#_Toc409064665)

[References 28](#_Toc409064666)

[Appendix 1 - PHY Abstraction 29](#_Toc409064667)

[Appendix 2 – Traffic model descriptions 31](#_Toc409064668)

[Appendix 3 – RBIR and AWGN PER Tables 41](#_Toc409064669)

[Appendix 4 – Packet Reception and Preamble Detection Procedure 41](#_Toc409064670)

# Revisions

|  |  |  |
| --- | --- | --- |
| **Revision** | **Comments** | **Date** |
| *R0* | Initial draft template | May 8th 2014 |
| *R1* |  | May 14th 2014 |
| *R2* | Update of MAC calibration and style template | May 15th 2014 |
| *R3* | Correction of traffic parameters in appendix 2, revised appendix 1, new VOIP models | July 17th 2014 |
| *R4* | Addition of energy efficiency metrics | September 18th 2014 |
| *R7* | Addition of Appendix 4 and revised RBIR description in Appendix 1 | January 15th 2015 |
| *R9* | Added text to clarify Box 2 MIMO simulation calibrations. Provides traffic model updates in support of mixed traffic modelling for each simulation scenario | May 14th 2015 |
| *R10* | Changed gaming traffic parameters |  |

# Notes on this version

In R2 added:

1. Text for box 3 of the calibration procedure
2. Variable bit rate coding heading in Appendix 2
3. Table of contents and style template

In R3 added:

1. Video traffic parameter correction
2. New text describing PHY abstraction in Appendix 1
3. New text for VOIP traffic

# Simulation Methodologies - General Concept

Two types of simulation methodologies are defined to enable the assessment of the performance and gain of proposed 11ax techniques relative to 11ac, each having its own advantages:

1. PER simulations – typically used for new PHY features for assessing point to point performance
2. System simulations – provide system-wise (multi-BSS, multi-STA) performance assessment with various degrees of detail as defined in the following three options:
   1. PHY system simulations – provide system-wise (multi-BSS) performance assessment with emphasis on PHY abstraction accuracy and very simplified MAC (e.g. transmissions are limited by CCA rules)
   2. MAC system simulations - provide system-wise (multi-BSS) performance assessment with emphasis on MAC accuracy and very simplified PHY (e.g. AWGN channel)
   3. Integrated system simulations – provide system-wise (multi-BSS) performance assessment with close-to-reality level of details accuracy by integrating both PHY and MAC

All three system simulation options have certain advantages and disadvantages:

1. Integrated system simulation:
   1. Provide comprehensive performance evaluation of PHY and MAC techniques in an environment that is close to a real-world scenario
   2. Provide deeper insight into PHY/MAC interworking:
      1. Techniques such as MU-MIMO or techniques for improving control frame delivery efficiency and reliability may require both PHY and MAC details.
      2. In some instances performance gain may only be revealed by observing the joint effects of both PHY and MAC models
      3. Enable the understanding of performance tradeoff between layers, e.g. some PHY rate enhancements may sacrifice MAC efficiency
2. PHY and MAC system simulations:
   1. Simplify some of the MAC/PHY details respectively
   2. Provide faster run time thus enabling more extensive research
   3. Speed up the project development by reducing dependency of PHY on MAC and vice versa
   4. Improve insight into the specific reason for performance gains/losses by isolating the MAC and PHY
   5. Enable accurate investigation of techniques that do not require all PHY/MAC details to be simulated

All system simulations options are used over the same simulation scenarios as defined in [10][11].

# System Simulation – High Level Description

A system simulation is comprised of multiple drops and multiple transmission events.

A drop is defined as a specific set of AP and STA locations within a topography. Different drops have different STA locations and possibly different AP locations as defined by the simulation scenario document [10] but the topography of the environment remains unchanged.

During a transmission event a set of transmissions occurs across multiple BSS. Multiple transmission events with typical aggregate duration 1-10[sec] beyond a warm-up time are required to assess the performance of a given configuration of APs and STAs. Each BSS may have different start time, duration and end time for its transmission event but time alignment (start, duration, end) of transmission events across different BSSs in the system is a possible outcome of a proposed MAC protocol.

A’warm-up’ period may be used to allow for some parameters to converge. For example:

1. MCS selection - if the MCS adaptation algorithm requires decisions based on past performance then the warm-up period may be used for initializing the algorithm.
2. Offered load - if all flows start exactly at T0, then the offered load goes from 0 to X instantaneously, and a high number of collisions will occur when there is a large number of STAs in the scenario. It will take a warm-up time for the system to recover to a stable operating condition.
   1. The backoff mechanism will effectively reduce the total offered load of the system by increasing the CW at each competing STA and thereby reducing its offered load, until the system total offered load is at Y < X

General simulation structure:

For drop=1:N {

Drop APs and STAs according to the description in [10]

Determine the channel for every link using distance-based PL, shadowing, wall/floor loss, and multipath model.

Associate STAs with APs according to the description in [10]

Note – determine users with SINR under that of MCS0 by ‘un-associated user’. Exclude un-associated users in evaluation. For the purpose of information, provide the percentage of un-associated users in evaluation

For transmission event=1:M {

* + Note – one can count time, ensuring that enough time has passed to see M transmission events
  + Note – the transmission event duration may not be the same in each BSS
  + Generate traffic at chosen nodes. Nodes chosen in compliance with
    - CCA rules and various other EDCA parameters
    - Channel access ordering rules (round robin, proportional fair, distributed access)
  + Generate packets consistent with a link adaptation algorithm
    - SU OL, SU BF, MU
    - MCS selection
  + Perform transmissions
  + Determine packet success or no
  + Collect metrics.

}

}

# PER Simulation Description

PHY PER simulations are used to verify point to point performance or aspects that are suitable for this type of simulation, such as new PHY features and preamble performance.

PHY impairments such as PA non-linearity, phase noise, synchronization error, channel estimation error and non-linear receivers are more readily incorporated into PER simulations and simulations that vary these parameters may be needed to test proposals if it is postulated that the techniques within those proposals are adversely affected by these impairments [6][9].

Other impairments such as the impact of OBSS interference or inter-symbol interference should also be verified by PER simulations by explicitly adding interfering packets to the simulation.

# PHY System Simulation Detailed Description

The emphasis here is on accurate modeling of the PHY using PHY abstraction (see description in Appendix I) with focus on DATA packets.

Only the very basic MAC is simulated. This is captured in the following description of a PHY system simulation using the approach taken in [17]:

1. Drop AP’s and STA’s according to scenario (random and/or deterministic placement)
   1. Ensure that every STA <-> associated AP link can sustain MCS0 (or another predetermined MCS) in both directions.
   2. Channel for every link in network determined by distance-based path loss, shadowing, wall/floor loss, and multipath model
      1. Independent shadowing for every TX-RX link
      2. Deterministic values for wall & floor loss
2. Once drop has been made, for link between every pair of devices in the building have:
   1. Path loss value, with path loss value accounting for shadowing and penetration losses
   2. Multipath channel
3. TX event: determine set of active TX nodes and RX SINR based on that set
   1. Initialize visited BSS set as empty.
   2. Randomly select an un-visited BSS
      1. Identify potential TX/RX pair in selected BSS: Randomly determine downlink/uplink according to downlink probability, and randomly select one of STA’s in selected BSS
      2. Check interference level from already activated TX’s at potential TX device
         1. Sum power in linear domain across interferers and tones, and average (in linear domain) across RX antennas to get aggregate interference
         2. If interference <= threshold, activate link and add potential TX to the set of already activated TX’s
         3. If interference > threshold, do not activate.
   3. Continue above until every BSS has been tried once.
   4. Once complete, the set of active TX nodes in the current TX event has been determined.
4. For each TX event, visit BSS’s in a random order -> thereby leading to possibly different active TX set for each TX event
5. For a single drop, run many TX events and compute a per-flow throughput
6. Flow is either uplink from a STA or downlink to a STA. Total # of flows = 2 \* # STA’s
7. Perform above across many drops to get averaging across spatial distribution

An implicit assumption is made that transmissions in OBSS are time synchronized since devices hear the preamble and defer for the duration of a packet.

# Integrated System Simulation Detailed Description

Integrated system simulation is a discrete-event simulation, which accurately models the behaviors of both PHY and MAC as a discrete sequence of events in time. Each event occurs at a particular instant in time and marks a change of state in the system. Between consecutive events, no change in the system is assumed to occur; thus the simulation can directly jump in time from one event to the next, as shown in Figure 1.



Figure 1: Clock advancement in an event-driven simulator

The feature set of integrated system simulation includes a minimal feature list and a nice-to-have feature list, as shown Table 1.

Table 1: Feature list of integrated system simulation

|  |  |  |
| --- | --- | --- |
|  | Full feature list | |
| Minimum features | Nice-to-have Features |
| MAC | CCA | Multiple channels |
| Control frame (RTS/CTS/ACK/Block ACK) | Control frame (CTS2self) |
| EDCA | Management frame |
| Aggregation (A-MPDU in 11ac) | … |
| Link Adaption |  |
| Transmission mode (SU-OL, Beamforming,…) selection |  |
| Power save mechanism (PS mode, PS polling, U-APSD |  |
| PHY | Beamforming vector | MU-MIMO |
| MMSE | … |
| Effective SINR Mapping and PER prediction |  |
| Energy detection |  |

MAC process should model the features of EDCA, CCA, aggregation, control frame (RTS/CTS/ACK) transmission and reception, link adaptation and sending the received result to statistics collection block, as illustrated in the Figure 2.



Figure 2: Detailed modelling of MAC

Notes: The feedback delay of channel state information in link adaptation should be considered.

PHY process includes abstraction of sending packets from MAC to channel, receive packets from channel and notify MAC. The following features should be detailed modeled, including beamforming vector, SINR calculation based on receiver algorithm, effective SINR mapping, PER prediction, energy detection, etc, as illustrated in Figure 3.



Figure 3: Detailed modelling of PHY

Integrated system simulation should follow the packet reception and preamble detection procedure as described in Appendix 4.

The simulation procedure follows the following steps:

## Step 1: initialization

* Drop APs and STAs according to description in [10], and initialise the internal state of each node device.
* Determine channel model for each AP and STA according to the description in [10].
* Associate STAs with APs according to description in [10].
* Create an event list as the main event scheduler of the simulator.

Notes: The location of each STA remains unchanged during a drop. Additionally, the STA is assumed to remain attached to the same AP for the duration of the drop.

## Step 2: event creation and processes

There are three types of events defined, including traffic generation event, MAC event, and PHY event. These events are inserted into the event list, and trigger subsequent MAC/PHY processes based on their particular time instant.

* Traffic generation event: is created by upper layer at the time instant of packet generation according to the traffic model. It triggers the packet generation process to generate a packet.

Note: the packet can include only the information of time instant and size, instead of actual bit stream.

* MAC event: is created by either upper layer at a transmitter or PHY layer at receiver. MAC events created by upper layer trigger the MAC process at the transmitter for the packet in MAC layer. MAC events created by PHY layer determine whether the packet is correctly received or not based on the PER predicted in PHY and trigger MAC process at the receiver when the packet is correctly received.
* PHY event: is created by MAC layer at a transmitter when the packet in MAC layer is ready for transmission. It triggers a PHY process at a receiver to predict PER for the packet.

Step 2 includes the following processes:

* packet generation process
  + For each traffic generation event, generate a packet including packet time instant and packet size
  + Create a MAC event when the packet is passed from upper layer to MAC layer
  + Create (next) traffic generation event according to each AP/STA’s traffic models

Notes: Start times for each traffic type for each STA should be randomized as specified in the traffic model being simulated.

MAC process at transmitter, if the MAC event is from upper layer:

* + Check CCA from energy detection in PHY and NAV in MAC
  + Carry out EDCA with CSMA/CA procedure
    - Count down backoff timer
    - Send RTS/CTS configurable by scenario/technique
  + Select transmission mode, e.g. SU OL, SU BF, MU, choose MCS, and perform packet aggregation, then create a PHY event and insert it into the event list based on the generation time of PHY event, and wait for PHY process
    - Packet aggregation rules specified in each simulation scenario are to be applied before transmission.

MAC process at receiver, if the MAC event is from PHY layer:

* + Determine the event success/failure based on PER as the abstract packet delivered by PHY
  + Send ACK/BA if packet transmission is successful
    - Notify the packet receive results to upper layer (Optional)

PHY process

* + Each AP/STA in the network performs energy detection and updates its CCA indication
  + Each AP/STA with channel busy in the network updates its NAV
  + TX: obtain precoding matrix, then notify RX
  + Channel: generate instantaneous fading channel (or load from offline files)
  + RX: calculate SINR of each tone based on receiver algorithms, e.g. MMSE, and perform PHY abstraction to obtain post SINR, and then PER
  + Create a MAC event to trigger MAC process at receiver

Repeat step 2 with sufficient simulation time to collect statistics.

## Step 3: Statistics collection

Collection the statistics according to the performance metrics defined in [x]

Note: in order to obtain reliable results, sufficient numbers of drops are simulated to ensure convergence.

Following is a more detailed description:

For drop=1:N

{

Step1:

{

Drop APs and STAs according to description in [10];

Associate STAs with APs according to description in [10];

Create event list for the scheduler of simulator;

Initialize the traffic generation event for each AP/STA;

}

Step2:

While simulation time is less than the end time

{

While traffic generation event occurs

{

Generate a packet of the size according to traffic model;

Create MAC event when the packet is passed to MAC;

Create the next traffic generation event at the time instant according to traffic model;

}

While MAC event occurs

{

If MAC event is from upper layer

{

If the CCA indicates idle and NAV is not set

{

EDCA with CSMA/CA procedure

{

Count down backoff timer;

}

Select transmission mode, e.g. SU OL, SU BF, MU;

Choose MCS;

Packet aggregation;

Create PHY events, and wait for PHY process;

}

}

If MAC event is from PHY layer

{

Determine the packet transmission success/failure based on PER;

If packet transmission is successful

{

Notify the packet receive status to upper layer (optional);

Send ACK;

}

}

}

While PHY event occurs

{

Each AP/STA in the network performs energy detection and updates its CCA indication;

Each AP/STA with channel busy in the network updates its NAV;

TX: obtain precoding matrix, then notify RX;

Channel: generate instantaneous fading channel (or load from offline files);

RX: calculate SINR of each tone based on receiver algorithms, perform PHY abstraction to obtain post SINR and get PER;

Create a MAC event to notify PER to MAC, and wait for MAC process;

}

}

Step3:

Collect statistics.

}

# MAC System Simulation Description

MAC system simulation is an integrated system simulation stripped out of the details of PHY modelling, e.g. a SISO configuration with AWGN - path loss and penetration loss should be modeled according to the scenario-specific definition.

MAC system simulation should follow the packet reception and preamble detection procedure as described in Appendix 4.

# Simulation Methodology Choice

Proponents of different techniques should provide justification for their proposed simulation methodology used to justify the technique’s gains. Proponents should also provide a comparison to performance with baseline parameters, e.g. .11ac.

Examples:

* PHY PER simulation:
  1. New PHY – a PER simulation is typically sufficient in order to decide the number of pilots, interleaver parameters and other parameters.
  2. Preamble performance
  3. Implementation losses of current and new PHY modes.
  4. Interference, especially if varying across the packet, impact on PER.
* PHY System simulation:
  1. Impact of number of antennas on multi-BSS performance
  2. Impact of PHY techniques in the context of multi-BSS
  3. Impact of frequency re-use in multi-BSS
  4. Impact of CCA levels on system throughput
* MAC System simulation:
  1. Impact of MAC scheduler – for example EDCA vs. RAW (as in 11ah) vs. HCCA vs. other techniques
  2. Impact of frequency re-use in multi-BSS
  3. Impact of CCA levels
* Integrated System simulation:
  1. Performance evaluation of 11AX solution in the environment close to real-world
  2. Impact of crosslayer techniques affecting both PHY and MAC layers in the context of multi-BSS

Note that some techniques can be simulated using multiple simulation tools to provide better insight

# System Simulation Calibration

Calibration of all system simulations is used to harmonize results between multiple entities and is depicted in the following flow chart whereby level of details is divided between several boxes with generally increasing level of detail starting at box 1 and ending in box 5.



Box 1 : Long-term statistics calibration

* The objective is to align the distribution of static radio characteristics.
* Static radio characteristics reflect the deployment, STA-AP association, and large-scale fading channel generation.

Box 2: multipath and MIMO are added

* The objective is to align distribution of accurate realistic channels (small and large scale fading) with MIMO configurations.

Box 3 : MAC system simulator calibration.

* The objective is to align the MAC system simulator using a defined set of features

Box 4: PHY system simulator

* The objective is to align the PHY system simulator.

Box 5: Integrated system simulator calibration

* The objective is to align a combination of all PHY and MAC features

Box 0: PHY abstraction is used in system simulations in lieu of running PER simulations

The following table specifies which box is used to calibrate which scenario. Box 1 is used for calibration of every scenario and only scenarios 1 and 4 are used for calibration of all boxes.

* Scenarios 2 and 3 have similar channels as scenario 1
* Scenario 3 has similar topology as scenario 4

Table 2: Box usage by scenario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Scenario 1 | Scenario 2 | Scenario 3 | Scenario 4 |
| Box 1 – long term SINR | Y | Y | Y | Y |
| Box 2 – multipath SINR | Y |  |  | Y |
| Box 3 – MAC SLS | Y |  |  | Y |
| Box 4 – PHY SLS | Y |  |  | Y |
| Box 5 – Integrated PHY/MAC SLS | Y |  |  | Y |

A detailed description of the calibration process is as follows:

## Box 1

The long-term SINR is defined as the ratio between the long-term received power from a desired transmitter and the sum of the long-term received power from all the interfering transmitters plus noise.

For example, if a STA is the transmitter, the intended receiver is the associated AP; if an AP is the transmitter, the intended receiver is one of the STAs associated with it. The interfering transmitters are defined in each test.

The long-term SINR of the receiver node-RX with the desired transmitter node-TX is defined as:



The summation of interference is over every BSS which contains at least 1 interfering transmitters. The number of interfering transmitter in the BSS that the receiver belongs to is always 0.

The long-term received power at a receiver node-RX from a (desired or interfering) transmitter node-TX is defined as:



### 

**Note 1:** For Box 1, there is no notion of an “un-associated user”, i.e., no check is performed to ensure that a STA can maintain MCS0.

### Note2: Transmit power is defined per-antenna as specified in [10]

### Test 1 (interference free)

No interfering transmitter is defined for each BSS. Therefore, for the n-th BSS, *Ω(k)={}* and *N(k)=0*, for *k=1…NBSS*.

Denote the set of *NSTA(k)* STAs associated with AP-k as *Ф(k)*. DL long-term SINR (received at each STA at all BSS’s) is explicitly defined as:



For this test, only DL long-term SINR is required.

### Test 2 (DL only)

For a receiver (STA) in the n-th BSS, the interfering transmitters are defined to be all non-associated APs. Therefore, *Ω(k)={AP-k}, N(k)=1* for *k=1…NBSS­, k≠n*.

DL long-term SINR (received at each STA at all BSS’s) is explicitly defined as:



### Test 3 (UL only)

Each BSS randomly selects a STA as the transmitter. For the AP in the n-th BSS, the interfering transmitters are defined to be all the activated and non-associated STAs. Therefore, *Ω(k)={STA-m, mϵФ(k)}, N(k)=1* for *k=1…NBSS­, k≠n*.

UL long-term SINR (received at each AP at all BSS’s) is explicitly defined as:



### Test 4 (DL/UL=1:1)

Each BSS randomly chooses DL or UL with p=0.5, and further randomly select a STA in this BSS. The transmitter and the receiver will be the AP and the selected STA depending on the DL/UL. For the receiver in the n-th BSS, the interfering transmitters are defined to be all the activated transmitters that are not in the same BSS. Therefore, , *Ω(k)={AP-k or STA-i, iϵФ(k)}*, *N(k)=1* for *k=1…NBSS­, k≠n*.

Long-term SINR (received at each AP or STA at all BSS’s) is explicitly defined as:



A separate CDF will be generated for DL and UL SINR.

### Test 5 (all nodes are active)

All the nodes are active. For each receiver AP or STA) in the n-th BSS, the interfering transmitters are defined to be all the APs and STAs in other BSS’s. Therefore, *Ω(k)={AP-k and STA-i, for all iϵФ(k)}*, *N(k)=NSTA(k)+1* for *k=1…NBSS­, k≠n*.

Long-term SINR (received at each AP or STA at all BSS’s) is explicitly defined as:



A separate CDF will be generated for DL and UL SINR.

### Procedure of test

* For each test on a selected calibration scenario and at least [x] drops of STA/AP is required for convergence.
* In each drop,
  + Drop STAs/APs, and associate each STA with an AP according to the scenario.
  + Activate the transmitter nodes and the receiver nodes, and collect the long-term SINR defined by the test.
* Generate the distribution (CDF) of long-term SINR collected over multiple drops.
  + The data format of SINR CDF is the SINR value for each percentile.
* Calibration goal:
  + The difference between distributions from multiple companies should be within [x]%.

## Box 2

The multipath SINR is defined as the instantaneous frequency-domain equalizer output SINR with the fading channels from both the desire transmitter and interfering transmitters.

For example, if a STA is the transmitter, the intended receiver is the associated AP; if an AP is the transmitter, the intended receiver is one of the STAs associated with it. The interfering transmitters are defined in each test.

The multipath SINR of the receiver node-RX with the desired transmitter node-TX at the m-th tone assumes the received signal as



The SINR depends on the receiver type (such as MRC/MMSE). The number of interfering transmitter in the BSS that the receiver belongs to is always 0.

For example, for SISO case (1x1, no precoding), the SINR is defined as



The long-term received power at a receiver node-RX from a (desired or interfering) transmitter node-TX is defined as in box1.

For MIMO, the SINRs of each spatial stream is collected for plotting results.

For example, for MIMO case with linear receiver, the SINR for j-th spatial stream is defined as

,

Where is the linear receive filter used at the receiver, represents the j-th column vector of the matrix, and  is the L-2 norm of a vector.

Note: For Box 2, there is no notion of an “un-associated user”, i.e., no check is performed to ensure that a STA can maintain MCS0.

### Test 1 (interference free)

For this test, only DL multipath SINR is required. No interfering transmitter is defined for each BSS. Therefore, for the n-th BSS, *Ω(k)={}* and *N(k)=0*, for *k=1…NBSS*.

### Test 2 (DL only)

For a receiver (STA) in the n-th BSS, the interfering transmitters are defined to be all non-associated APs. Therefore, *Ω(k)={AP-k}, N(k)=1* for *k=1…NBSS­, k≠n*.

### Test 3 (all nodes per channel access rule)

A channel access rule is used to select 0 or 1 transmitters per BSS. For the corresponding receiver in the n-th BSS, the interfering transmitters are defined to be the nodes that obtain channel access by the rule. Therefore, *Ω(k)={none, or AP-k, or STA-i, for some iϵФ(k)}*, *N(k) = 0 or 1*, for *k=1…NBSS­*.

Channel access rules use CCA are defined as:

1. Order all nodes (STAs and APs) and put them in a list in a random order, initialize a set of transmitter *T*={}, and *Ω(k)={}*, for *k=1…NBSS­*
2. While the list is not empty:
   1. Select the first node and remove it from the list.
   2. Calculate the interference based on the current *Ω(k),* for *k=1…NBSS­.*That is,



The interference power is measured across the entire simulation bandwidth.

* 1. Compare the interference with a CCA threshold of -70dBm. If the interference is smaller than the threshold,
     1. Add this node to *T*, and set *Ω(n)={this node},*if this node belongs to BSS-n.
     2. If this node is an AP, randomly selects an associated STA as receiver; if this node is a STA, the receiver is the associated AP.
     3. Remove all existing nodes in the same BSS from the list.

1. Output the set of transmitter *T*.

A flow chart of such a CCA-based channel access rule:



Figure 4: CCA-based channel access rule flow chart

A separate CDF will be generated for DL and UL SINR.

### Procedure of test

* For each test on a selected calibration scenario, at least [x] drops of STA/AP and [x] TX events per drop are required for convergence.
* In each drop,
  + Drop STAs/APs, and associate each STA with an AP according to the scenario.
  + In each TX event, select the transmitter and receiver nodes, and collect the multipath SINR per tone for the pairs of transmitter/receiver.
  + The fading channel evolves over the TX events (detailed to be added).
* Generate the distribution (CDF) of multipath SINR collected over the simulation time from multiple drops.
  + The CDF of SINR per tone.
  + The CDF of effective SINR per reception based on capacity mapping, i.e.,



where *Ntones* is the number of tone and NSTS is the number of spatial streams.

.

* + The data format of SINR CDF is the SINR value for each percentile.
* Calibration goal:
  + The difference between distributions from multiple companies should be within [x]%.

## Box 0

The calibration is repeated for BCC and LDPC coding schemes:

* For packet length *PL*, estimating *PERPL* from following equation*PERPL* = 1-(1-*PERPL*0)*PL*/*PL*0
* In case of BCC, *PL*0 is 32bytes for less than 400bytes and 1458bytes for other sizes
* In case of LDPC, *PL*0 is 1458bytes for all packet sizes

Assume ideal channel estimation (The impact of practical CE error will be counted as additional noise in per-tone SINR calculation in system simulation).

An L-MMSE (equivalently MRC for single spatial stream) receiver is assumed.

### Step 1: Align AWGN link performance

* For each MCS (MCS0 to MCS9), provide SNR vs. PER curves at a SNR step size of 2dB.

### Step 2: Verify the effective SNR vs. PER performance against AWGN results

* Specify the effective SNR mapping method used (e.g., RBIR/MMIB/EESM/Constrained Capacity) and tuning parameters (if any).
* Simulate over 11nB\_NLOS, 11nD\_NLOS
  + For each channel type:
    - Simulate over a range of SNR in 1dB steps down to 1% of PER for each MCS
    - For each SNR, simulate over at least 5000 independent channel realizations
      * For each channel realization collect the effective SNR (a scalar for one spatial stream) and 1-bit flag of decoding result (success or failure) 🡪 a 1x2 vector as output [SNR\_eff, flag]
      * Combine the large collections of [SNR\_eff, flag] over all realizations and SNRs, and quantize to effective SNR in 0.25dB steps vs. PER table
      * PER = (# of successful decoded packets in a SNR\_eff bin) / (# of packets in this SNR\_eff bin)

### Step 3: Verify the SNR vs. PER performance against fading channel results

* Specify the effective SNR mapping method used, tuning parameters, and the effective SNR vs. PER curves obtained in Step 2.
* Simulate over 11nB\_NLOS, 11nD\_NLOS
  + The channel realizations are generated independently of step 2.
  + For each channel type:
    - Simulate over a range of SNR in 1dB steps down to 1% of PER for each MCS
    - For each SNR, simulate over at least 100 independent channel realizations.
    - For each realization run at least 1000 packets and for each packet decide if it has been successfully received by the prediction-based or simulation-based methods.
      * For the predicted decision, compute the effective SINR, and find the *predicted* PER. Decide that this packet is successfully decoded if a random variable drawn uniformly in [0,1] is larger than the predicted PER.
      * For the simulated decision, determine that this packet is successfully decoded if CRC check is passed at the output of decoder.
      * Calculate the PER for this SNR value by PER = (# of successful decoded packets at this SNR level) / (# of packets for this SNR level)
      * Notes:
        1. The numbers of successful decoded packets are counted for both prediction based on PHY abstraction and decoder output.
        2. PER is quantized by the received SNR level, instead of effective SNR levels as in step 2.
      * PER is considered to be accurately predicted if the standard deviation of the SNR gap at the 10% and 1% SNR points is less the [TBD]dB

## Box 3

To improve the validity and credibility of MAC simulator, two types of testing are proposed, feature tests, and performance tests. Feature test is mainly designed to check the core MAC functions and should be PHY agnostic.  Core MAC functions include EDCA, BlockAck and A-MPDU. Feature tests are mandatory.   Performance tests are added to help verify some additional MAC functionality and these tests are optional.

Feature tests are designed to have expected outputs which should include various traces. Traces may include event timing traces, MAC state traces, sequence numbers, buffer contents, etc. Critical check points can be identified within these traces and can serve as an indicator of correct MAC behavior per IEEE specification. Feature tests can also test metrics such at Tput. On the other hand, performance tests collect the statistics of the system performance for the complete simulation scenarios described in [10].

It is required to calibrate individual MAC features via the following set of   simplified scenario tests, (see [10] in scenarios for calibration of MAC simulator)

* + Test 1: Overhead Tests
    - MAC overhead w/o RTS/CTS
    - MAC overhead w/ RTS/CTS
  + Test 2:  Deferral tests.
    - Test that APs defer when they should
    - Test that APs don’t defer when they shouldn’t
  + Test 3: NAV test
  + More tests to be added

If planning on using the MAC simulator as a standalone simulator to bring in simulation results, tests using the full simulation scenarios as defined in [10] would be needed.

* + In this case, the simulation parameters to be used are defined in the scenario itself.
  + Note that certain behavior not explicitly defined by the standard may be implemented differently by different companies; each test should also clarify the assumption for the modeling of relevant behavior not defined by the standard.

## Box 4

Box 4 is based on the process described in Box 2 test 3 and augmented to include MCS and PER as described in box 0.

MCS choice is based on:

* Genie; or
* Goodput maximizing MCS

The output metrics are CDF of:

* Per-STA throughput
* SINR of active links
* Per-STA selected MCS
* Per-STA airtime

## Box 5

Box 5 should calibrate the system level performance in the scenario calibrated in box 1 and box 2 based on the MAC and PHY features listed in the following table based on the results of box 3 and box 4.

Table 3: MAC and PHY features

|  |  |
| --- | --- |
| **Feature** |  |
| MAC | CCA |
| Control frame, e.g. RTS/CTS/ACK/Block ACK |
| EDCA |
| Aggregation (A-MPDU in 11ac) |
| Link Adaption |
| PHY | Beamforming |
| MIMO with MMSE receiver |
| PHY abstraction (see more details in Appendix 1) |

Basic packet reception and preamble detection procedure are described in Appendix 4 with the following simplifications for calibration,

* + - For control frame, the preamble only includes the legacy part
    - For data frame, both legacy part and VHT part are included.
    - Link adaptation and channel estimation model are ignored unless otherwise stated
    - The duplicated parts in preamble among multiple 20MHz channels are not combined
    - The receiver will be locked by the first-arrived packet, and later-arrived packets are considered as interference.
    - The size of the preemption window is set to 0 or the value of P dB approaches infinity, i.e., if the preamble of one PPDU is decoded successfully, no other PPDUs will be detected through the end of the PPDU.
    - MCS0 is assumed for control frame
    - Rx sensitivity is assumed to be equal to the value of CCA-SD.

Test 1: full buffer traffic model assumption

Test 2: real traffic model and traffic mix assumption defined in evaluation methodology and simulation scenario document.

The output metrics are CDF of :

* Per-STA throughput
* Per-BSS Throughput
* Packet Loss
* Transmission Latency

# Traffic Models

Full buffer model is baseline – users always have DATA to send and receive.

A more realistic FTP traffic model may be used based on [15]. Specific parameters are TBD.

A mix of small and large packets should be evaluated in order to test realistic assumptions on system performance.

Traffic models for Video are described in Appendix 2.

## Management traffic model

Management traffic model for unassociated clients:

* Probing period:
  + For {50%} of the clients: [12 seconds]
  + For {50%} of the clients:
    - [12.5 seconds]
    - If still unassociated after [5] times probing all the channels, then probe all the channels with doubled Probing period, and maximum period of [400 seconds].
* Probing channels: Every supported channel [1,2,3,4..,36,40,..]
* Probe request SSID: Broadcast probe requests to wildcard SSID, plus [0-3] specified SSIDs
* Probe Request frame size: [80B, or 160B]

Management traffic model for associated clients:

* Probing period: [60 seconds]
* Probing channels: Same channel that the client is associated, unless the associated AP Beacon’s RSSI is below [TBD dBm] in which case probe every supported channel [1,2,3,4..,36,40,..]
* Probe Request frame size: [80B, or 160B]

Probe request SSID: Probe the associated AP/SSID if RSSI is not below [TBD dBm], otherwise broadcast probe requests to wildcard SSID

# Metrics

11AX evaluation methodology defines evaluation of spectrum efficiency improvement in both link level and system level.

## Link Level Simulation

For PER simulations the typical metric is dB gain/loss in waterfall curves. The operating range to be observed is 1% to 10% PER.

## System Level Simulation

For system simulations it is suggested to use the following metrics to evaluate the system performance [2]-[9], [19]-[21]:

### Per-STA Throughout

Per-STA throughput metrics are used to measure the user experience in the area covered by one or multiple BSSs in different simulation scenario [10].

Definition – Per-STA throughput is measured at MAC SAP by the number bits (or bytes) of MAC payload successfully transmitted over the given measurement period in the full buffer simulation.

• Per-STA throughput at 5 percentile of throughput CDF curve measures the minimum throughput performance of stations at the cell edge.

• Per-STA throughput at 50 percentile of CDF curve measures the average throughput of stations in all participating BSS in the simulation.

• Per-STA throughput at 95 percentile of CDF curve measures the top performance of stations at the cell center of BSS.

Although the main target of 11AX is to improve the performance at 5 and 50 percentile of throughput CDF curve, it is suggested to measure Per-STA throughput at the 5, 50, and 95 percentile points. The entire throughput CDF curve and other information such as MCS histogram may help to evaluate the overall system performance improvement [3].

Per-STA throughout for DL and UL are measured separately.

### Per-BSS Throughput

Per-BSS throughput is used to evaluate BSS capacity in the various simulation scenarios described in [10]. This metric directly relates to the aggregated Per-STA throughputs in BSS and can be used to compare different deployment densities and heterogeneous deployments.

Definition – Per-BSS throughput is the aggregated Per-STA throughput among all the associated stations in a BSS.

Per-BSS throughout could be measured by aggregating Per-STA throughputs of all the stations in a BSS, or derived from Per-STA throughput times the number of associated stations in a BSS.

Per-BSS throughout for DL and UL are measured or calculated separately.

### Packet Loss

The packet loss metric is used to evaluate the system robustness especially in the high density deployment scenario. This metric reflects an aspect of system performance different from throughput and transmission latency.

Definition – The packet loss is defined as the number of MAC packet not delivered at all or not delivered in time to the receiver over the total number of offered MAC payloads.

The packet loss means that the MAC packet could not be decoded by the receiver due to the interference or low RSSI, or the MAC packet could not be delivered at the receiver in time for QoS flow due to traffic congestion.

### Transmission Latency

The metric of transmission latency is used to measure the time delay of medium acquisition in channel access mechanism. The transmission latency is used to evaluate an aspect of MAC performance in various QoS transmissions.

Definition – The transmission latency is measured from the time that MAC receives a packet till the time that PHY starts transmitting.

The transmission latency may include the time delay of

• AIFS

• Backoff time

• Other system parameters

**Per STA Energy per Transmit Bit**

The metric of per STA energy per transmitted bit, measured in units of joules per bit, is defined as the total energy consumed by a STA divided by the total number of successful data bits transmitted by the STA.

**Per STA Energy per Receive Bit**

The metric of per STA energy per received bit, measured in units of joules per bit, is defined as the total energy consumed by a STA divided by the total number of successful data bits received by the STA.

**Energy Efficiency Ratio (EER)**

Energy efficiency ratio is defined as the ratio of average energy consumed during one successfully exchanged data bit between STAs using any new proposed power save mechanism over the baseline power save mechanism.



**Network Energy Efficiency Ratio (N-EER)**

N-EER defined as the ratio of average energy consumed for M stations of interests during an event (Tevent) for series of successfully exchanged data bits between STAs using any new proposed power save mechanism over the baseline power save mechanism. The mathematical calculation of N-EER is described as follow.



The values for voltage and current may be chosen from Power Model Parameter table in Simulation Scenario document [26]

# References

[1] 11-13-0657-02-0hew-hew-sg-usage-models-and-requirements-liaison-with-wfa

[2] 11-13-0486-01-0hew-metrics-targets

[3] 11-13-0847-01-0hew-evaluation-criteria-and-simulation-scenarios

[4] 11-13-0869-00-0hew-simulation-scenarios-and-metrics-for-hew

[5] 11-13-0850-00-0hew-quantitative-qoe-requirements-for-hew

[6] 11-13-0722-01-0hew-hew-evaluation-methodology

[7] 11-13-0723-00-0hew-hew-sg-evaluation-methodology-overview

[8] 11-13-0786-00-0hew-hew-sls-methodology

[9] 11-13-0837-00-0hew-considerations-on-hew-evaluation-methodology

[10] 11-14-0621-05-00ax-simulation-scenarios

[11] 11-13-1001-06-0hew-simulation-scenarios-document-template

[12] IEEE 802.16m-08/004r5, IEEE 802.16m Evaluation Methodology Document (EMD) Section 4 [http://ieee802.org/16/tgm/core.html#08\_004](http://ieee802.org/16/tgm/core.html)

[13] 11-13-1059-00-0hew-PHY abstraction for HEW evaluation methodology

[14] 11-13-1131-00-0hew-PHY abstraction for HEW system level simulation

[15] 3GPP TR 36.814 Annex A.2.1.3.1 FTP Traffic model 1

[16] 11-13-1334-00-0hew-video-traffic-modeling

[17] 11-14-0082-00-0hew-Improved-Spatial-Reuse-Feasibility-Part-I

[18] 11-14-0083-00-0hew-Improved-Spatial-Reuse-Feasibility-Part-II

[19] 11-14-0107-00-0hew-hew-evaluation-metrics

[20] 11-14-0101-02-0hew-coments-on-802-11-hew-draft-par-5c

[21] 11-13-0805-02-0hew-on-definition-of-dense-networks-and-performance-metric

[22] 11-14-0353-00-0hew-suggestion-on-phy-abstraction-for-evaluation-methodology

[23] 11-14-1161-03-00ax-parameters-for-power-save-mechanisms

[24] 11-14-1162-01-00ax-energy-efficiency-evaluation-methodology-follow-up

[25] 11-14-1523-05-00ax-offline-discussion-minutes-of-sls-calibration

# Appendix 1 - PHY Abstraction

The objective of PHY abstraction is to accurately predict PER simulation results in a computationally efficient way to enable running system simulations in a timely manner.

The underlying principle is to calculate an effective average SINR (*SINReff* ) in a given OFDM symbol. This quantity then acts as a link between AWGN PER and multipath channel PER for a given coding type, block size and MCS level.

Effective SINR (*SINReff* ) is typically calculated as follows



where *SINRn* is the post processing SINR at the *n*-th subcarrier, *N* is the number of subcarriers in a coded block and Φ is a mapping function.

For PHY and integrated system simulation, the mapping function is based on received bit mutual information rate (RBIR) assuming coded-modulation. For M-QAM, the mapping function is defined as



where *U* is a zero-mean complex Gaussian random variable with variance 1, i.e., *U~CN(0,1)*. The lookup tables of Φ functions for different modulations are attached in appendix 3 with an SNR granularity of 0.02dB (note that high granularity is needed to accurately capture the steepness of the curve).

A lookup table of AWGN PER vs. SNR for each MCS and each coding scheme (BCC, LDPC) is collected with the SNR granularity of 0.1dB for 2 reference packet lengths for BCC and 1 reference packet length for LDPC. The AWGN PER lookup tables are attached in Appendix 3.

To determine if a PPDU of *Nss* spatial streams over *N* tones has been successfully received, a RBIR-based PHY abstraction is illustrated in the following diagram.



Figure 5. RBIR based PHY Abstraction Procedure

The detailed procedure of RBIR based PHY abstraction as shown in Figure 5 is:

Assume *K* interference events happen during a subframe of *L* OFDM symbols corresponding to a MPDU of *PL* bytes, each starts from *Tk* and lasts *Lk* OFDM symbols, for *k=1…K*.

For each interference event over *[Tk , Tk+1),* *k=1…K*:

1. Generate (both desired and interfering) channels.
2. Calculate the equalizer-output SINR per spatial stream for the *n*-th tone/*t*-th OFDM symbol, *SINR(iss,n,t), iss=1…Nss, n=1…N, t=Tk\*.*
   1. Equalizer is MRC if *Nss=1*, or MMSE if *Nss>1*.
   2. SINR is only calculated over 1 OFDM symbol for each interference event. In the case that *Tk* is not aligned with the OFDM symbols boundary of the signal to be decoded, use the first OFDM symbol after *Tk* for SINR calculation. That is,



1. Map *N*×*Nss* SINRs to *1* RBIR.



1. Reverse map *1* RBIR to *1* effective SNR.



1. Estimate PER for based on AWGN PER lookup table.



* 1. In case of BCC, *PL*0 = 32bytes if *PL*< 400bytes; or 1458bytes otherwise.
  2. In case of LDPC, *PL*0 = 1458bytes.

After iterated over all events,

1. Calculate the final PER



1. Determine if this subframe is successfully received.



# Appendix 2 – Traffic model descriptions

## Wireless Display (lightly compressed video) Traffic Model

Wireless display is a single-hop unidirectional (e.g., laptop to monitor) video application. The video slices (assuming a slice is a row of macro blocks) are generated at fixed slice interval. For example, for 1080p, the slice interval is 1/4080 seconds.

The video slices are typically packetized into MPEG-TS packets in wireless display application. But for 11AX simulation, we will ignore the MPEG-TS packetization process and assume video slices are delivered to MAC layer for transmission directly.

The traffic model for wireless display is modified from [TGad] with modifications below due to the fact that some parameters have dependency on video formats.

1. Parameters
   1. Set **IAT**, **MaxSliceSize** according to video format as Table 4.
   2. Normal distribution parameters
      1. µ = 9.19 Kbytes
      2. σ = 1.350 Kbytes
      3. b = 300 Mbps
2. Algorithm for generating each video slice/packet

* Input: target bit rate in Mbps (**p**)
* Output: slice size in Kbytes (L): At each IAT, generate a slice size L with the following distribution: Normal(µ\*(p/b), σ\*(p/b))
  + - If L > MaxSliceSize, set L= MaxSliceSize

Table 4: Model parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Traffic Model Class Identifier** | **Video format** | **Inter-arrival time (IAT)** | **MaxSliceSize** | **p** |
| WD1 | 1080p60 | 1/4080 seconds | 92.160 Kbytes | 300 |
| WD2 | 1080p60 3D | 1/4080 seconds | 92.160 Kbytes | 450 |
| WD3 | 4K UHD (3840x2160) 60fps | 1/8100 seconds | 184.320 Kbytes | 600 |
| WD4 | 8K UHD (7680x4320) 60fps | 1/16200 seconds | 368.640 Kbytes | 1200 |

Note: the data rate increase from 1080p to higher resolution is not linearly scaling as the uncompressed data rate due to higher redundancy in the images at higher resolution. Similar argument applies to 3D video. A 100% increase is assumed for 4K video as compared to 1080p, and 50% bit rate increase for 3D from 2D video.

### Evaluation metric

* MAC throughput, latency

## Buffered Video Steaming (e.g., YouTube, Netflix) Traffic Model

Unlike wireless display, video streaming is generated from a video server, and traverses multiple hops in the internet before arriving at AP for transmission to STA. It is a unidirectional traffic from the video server to the station, with reciprocal TCP Acknowledgement of the video frames.

Typically, Video streaming application runs over TCP/IP protocol, and video frames will be fragmented at TCP layer before leaving the video server. Since these TCP/IP packets experiences different processing and queuing delay at routers, the inter-arrival time between these TCP/IP packets are not a constant despite the fact that video frames are generated at constant interval at the video application layer.

### STA Layering Model

STA layering model is shown in Figure xx. Both AP and STA generate video frames at application layer. The video traffic goes through TCP/IP layer and then to MAC layer. The TCP protocol used for video streaming simulation is the same as other traffic model.



Figure 5: Traffic layering model

### Video traffic generation

The video traffic from Video Source to Video Receiver is generated as follows.

**Step 1**: At application layer, generate video frame size (bytes) according to Weibull distribution with the following PDF.


f(x;\lambda,k) =
\begin{cases}
\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1}e^{-(x/\lambda)^{k}} & x\geq0 ,\\
0 & x<0,
\end{cases}

Depending on the video bit rate, the parameters to use are specified in Table 5.

Table 5: Lambda and k parameter for video bit rate

|  |  |  |  |
| --- | --- | --- | --- |
| **Traffic Model Class Identifier** | **Video bit rate** | **lambda** | **k** |
| BV1 | 2Mbps | 6950 | 0.8099 |
| BV2 | 4Mbps | 13900 | 0.8099 |
| BV3 | 6Mbps | 20850 | 0.8099 |
| BV4 | 8Mbps | 27800 | 0.8099 |
| BV5 | 10Mbps | 34750 | 0.8099 |
| BV6 | 15.6 Mpbs | 54210 | 0.8099 |

**Step 2**: AT TCP layer, set TCP segment as 1500 bytes and fragment video packet into TCP segments.

**Step 3**: Add network latency to TCP/IP packets when these segments arrive at AP for transmission. The network latency is generated according to Gamma distribution whose PDF is shown below

f(x;k,\theta) =  \frac{x^{k-1}e^{-\frac{x}{\theta}}}{\theta^k\Gamma(k)} \quad \text{ for } x > 0 \text{ and } k, \theta > 0.

Where

* + k=0.2463
  + theta=60.227

The mean of the latency with the above parameters is 14.834ms. To simulate longer or shorter network latency, scale theta linearly since mean of Gamma distribution is K\*theta

If network latency value is such that the packet arrives at MAC layer after the end of the simulation time, then re-generate another network latency value until the packet arrives at MAC within the simulation window.

The reciprocal TCP Ack to the video traffic, from Video Receiver to Video Source is generated as follows:

Step 1: 40 Byte TCP Ack, inter-arrival interval is equal to 1ms delay from video traffic frame reception at the Video Receiver

### Evaluation metrics

* MAC throughput, latency
* TCP throughput, latency

## Video Conferencing (e.g., Lync) Traffic Model

Unlike buffered video streaming where video traffic is unidirectional and heavily buffered at the receiver, video conferencing is bi-directional video traffic with limited tolerance for latency. Video traffic is generated at each station, sent to AP, traverses the network/internet, reaches another AP, and then is transmitted to its destination STA.

### Station layer model



Figure 6: Video conferencing model

Because the traffic from AP to station has experienced network jitter, it can be modelled the same way as the traffic model of video streaming.

For traffic sent from Station to AP, since the traffic has not experienced network jitter, it is a periodic traffic generation as the first two steps described in video streaming.

### Video traffic generation

Traffic model from AP to station: use the same model as video streaming BV1.

Traffic model from station to AP: use the first two steps in video streaming traffic model BV1

Traffic Model Class Identifier for Video Conferencing is VC

### Evaluation metrics

* MAC throughput, latency

## Application event models

Application event model is used to specify the patterns of the application events, i.e., when to start the applications and how long for each application in the simulation. Different use scenarios may choose different application event models in the simulation.

* Poisson model

Poisson model can be used for random application event pattern where there are many users, each generating a little bit of traffic and requesting network access randomly.

Parameters: TBD

* Hyper-exponential model

Hyper-exponential model can be used for peak event pattern where users requesting network access in big spikes from the mean.

Parameters: TBD

## Multicast Video Streaming Traffic Model

Multicast Video Streaming is unidirectional video traffic from AP to multiple (1 or more) STAs.

The video traffic is generated from a video server, and traverses multiple hops in the internet before arriving at AP for transmission to STAs.

### Station layer model

****

Figure 7: Multicast streaming model

AP generates video frames at application layer.

Because the traffic from AP to stations has experienced network jitter,

it can be modelled the same way as the traffic model of video streaming.

The video traffic goes through UDP/IP layer and then to MAC layer. The video traffic is transmitted in the MAC layer as multicast RA addressed frames, NOT multiple duplicate instances of unicast RA addressed frames.

### Video traffic generation

Traffic model from AP to station: use the same steps in video streaming traffic model

We assume bit rate for video streaming 6 Mbps (1080/30p AVC) and 3 Mbps (1080/30p HEVC)

|  |  |  |  |
| --- | --- | --- | --- |
| **Traffic Model Class Identifier** | **Video bit rate** | **Lamda** | **K** |
| MC1 | 3Mbps | 10425 | 0.8099 |
| MC2 | 6Mbps | 20850 | 0.8099 |

**Evaluation metrics**

MAC throughput, latency

## Gaming Traffic Model

First Person Shooter (FPS) is a typical representative game of Massively Multiplayer Online (MMO) game. The FPS traffic model is considered to be a typical gaming traffic model, as it has additional requirements on, for instance, real time delay with irregular traffic arrivals. Gaming is asymmetric, bi-directional single-hop video traffic.

Gaming traffic can be modelled by the Largest Extreme Value distribution. The starting time of a network gaming mobile is uniformly distributed between 0 and 40 ms to simulate the random timing relationship between client traffic packet arrival and reverse link frame boundary. The parameters of initial packet arrival time, the packet inter arrival time, and the packet sizes are illustrated in the Table 6 [13]:

Table 6: Parameters for gaming traffic model

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Component** | **Distribution** | | **Parameters** | | **PDF** |
| **DL** | **UL** | **DL** | **UL** |
| Initial packet arrival (ms) | Uniform | Uniform | a=0,  b=20 | a=0,  b=20 |  |
| Packet arrival time (ms) | Largest Extreme Value | Largest Extreme Value | a=15,  b=7 | a=23.5,  b=10.5 |  |
| Packet size (Byte) | Largest Extreme Value | Largest Extreme Value | a=390,  b=89 | a=158,  b=26.2 |  |

\* A compressed UDP header of 2 bytes and a IPv4 header of 20 bytes (if use IPv6 here, the header should be 40bytes) has been accounted for in the packet size.

Traffic Model Class Identifier for Gaming is GMG

### Evaluation metrics

MAC throughput, latency

## Virtual Desktop Infrastructure Traffic Model

Virtual desktop infrastructure (VDI) traffic is generated from a server, and traverses multiple hops in the intranet before arriving at AP for transmission to STA. For the transmission from AP to STA, it is asymmetric, bidirectional single-hop traffic between AP and STA. VDI traffic transfers from server to STA/client via AP over TCP/IP protocol. This model describes the attribution of traffic from AP to STA, and VDI application type navigation and feedback traffic from the STA to AP

The VDI traffic from AP to STA is generated as follows:

**Step 1:** VDI traffic generation

The VDI traffic is generated as shown in Figure xx. At MAC layer, arrival interval of VDI packets is generated according to exponential distribution.



Figure 8: Traffic generation model

Traffic direction specific parameters for packet arrival time are specified in Table 7.

**Step 2**: At MAC layer generate VDI MSDU frame size (in bytes) for uplink and downlink transmission, respectively.

For uplink the packet size is generated according to a Normal distribution. For downlink the packet size is generated with a bimodal Normal distribution. The traffic direction specific PDFs and the packet size parameters are specified in Table 7.

Table 7: Parameters for VDI traffic model

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Component** | **Distribution** | | **Parameters** | | **PDF** |
| **DL** | **UL** | **DL** | **UL** |
| Initial packet arrival (ms) | Uniform | Uniform | a=0,  b=20 | a=0,  b=20 |  |
| Packet arrival time (ms) | Exponential | Exponential |  |  |  |
| Packet size (Byte) | Bimodal Normal | Normal |  |  |  |

Traffic Model Class Identifier for VDI is VDI

### Evaluation metrics

* MAC throughput
* Latency

**Voice-over-IP (VoIP) Traffic Model**

VoIP service uses the internet protocols to deliver real-time voice packets across networks. VoIP traffic is symmetric, bi-directional between AP and STA. The VoIP traffic comprises periods of active talking and silence, as shown in Figure 1. It can be considered as a simple 2-state noice activity Markov model as shown in Figure 2



**Figure 1: VoIP traffic profile**



**Figure 2: Two-state voice activeity model**

For VoIP traffic, the VoIP user will always be in either the silence state (State 0) or active talking state (State 1), assuming that the probability of transitioning from state 0 to state 1 is *a*, and the reciprocal transition from state 1 to state 0 is *b*. Hence, the probability of staying in state 0 is 1*-a* and in state 1 is 1-*b*. The state update is assumed to be done at the speech encoder frame rate *R*=1/*T*, where *T* is the encoder frame duration whose typical value is 20ms for active talking state and 160ms for silence state, respectively. VoIP packets are generated at time intervals *iT*+*τ*, where *τ* is the network packet arrival delay jitter, and *i* is the encoder frame index. During the active state, voice packets with fixed size are generated at these time intervals, while the model is updated at regular frame intervals [1].

The detailed parameters of the VoIP traffic model are specified in Table 1. The rate of voice source assumes 12.2 kbps with a 50% voice activity factor. The payload size of active talking state and silence state are 33 byte and 7 byte respectively. Compressed protocol headers including UDP check sum are used in the traffic, which is 3 byte for IPv4 and 5 byte for IPv6. The total voice MSDU frame sizes for active talking state are 36 byte and 38 byte for IPv4 and IPv6 respectively, and for silence state are 10 byte and 12 byte for IPv4 and IPv6 respectively.

Table 1: Parameters for VoIP traffic model

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Distribution** | **Parameters** | **PDF** |
| Source rate | N/A | 12.2 Kbps | N/A |
| Active packet payload size | N/A | 33 byte | N/A |
| Silence packet payload size | N/A | 7 byte | N/A |
| Compressed protocol headers | N/A | IPv4: 3 byte  IPv6: 5 byte | N/A |
| Voice encoder interval | N/A | 20+τ ms | N/A |
| Noise encoder interval | N/A | 160+τ ms | N/A |
| Active/Silence state duration | Exponential | Mean=1.25 second |  |
| Downlink delay jitter | Laplacian | *β*=5.11 ms |  |
| Uplink delay jitter | N/A | 0 | N/A |
| Voice activity factor | N/A |  | N/A |
| a | N/A | 0.016 | N/A |
| b | N/A | 0.016 | N/A |

Traffic Model Class Identifier for VoIP is VOIP

**Evaluation metrics**

MAC throughput, latency

**Local File Transfer (FTP) Traffic Model**

FTP, File Transfer Protocol is a standard network protocol used to transfer computer files from an FTP Source to an FTP Client over a TCP-based network, such as the Internet. FTP traffic is asymmetric, bi-directional with large, fixed-size block data frames in one direction, from an FTP Source to an FTP Client, and TCP ACK responses in the other direction, from the FTP Client to an FTP Source. FTP traffic is modeled as a sequence of file transfers separated by reading time, where reading time is defined as the time between the end of file transmission and the start of the subsequent file transmission. The packet call size is equivalent to the file size (S) and the packet call inter-arrival time is the reading time (D). A typical FTP traffic pattern is shown in Figure 1



**Figure 1 FTP traffic pattern**

The FTP Source traffic generation process is, at first, to create a file using the file size statistics in Table 1 which provides the model parameters for FTP traffic that includes both DL and UL, with either MTU size is 1500 bytes or 576bytes, and then to complete the transfer of the file using a new TCP connection with initial window size W=1, eventually waiting for a reading time until next file transfer.

Based on the results on packet size distribution, 76% of the files are transferred using an MTU size of 1500 bytes and 24% of the files are transferred using an MTU size of 576 bytes. Note that these two packet sizes calculated from the statistical distributions in Table 1 also include a 40 byte IP packet header.

**Table 1: Parameters for FTP traffic model**

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Distribution | Parameters | PDF |
| File  size (S) | Truncated Logonormal | Mean = 2 Mbytes  SD = 0.722 Mbytes  Max = 5 Mbytes | if x > max or x < min, discard and generate a new value for x |
| Reading  time (D) | Exponential | Mean = 180 second |  |

The reciprocal TCP Ack to the FTP traffic, from FTP Client to FTP Source is generated as follows:

Step 1: 40 Byte TCP Ack, inter-arrival interval is equal to 1ms delay from FTP traffic frame reception at the FTP Client

Traffic Model Class Identifier for FTP is FTP

**Evaluation Metrics**

* MAC Throughput, latency

**Web Browsing (HTTP) Traffic Model**

HTTP traffic is governed by the structure of the web pages on the World Wide Web (WWW), and commonly has a bursty profile due to the characteristics of human interaction, where the HTTP traffic pattern is as shown in Figure 2.

HTTP traffic is asymmetric, bi-directional with large, fixed-size block data frames in one direction, from an HTTP Server to an HTTP Client, and HTTP request packet(s) and TCP ACK responses in the other direction, from the HTTP Client to an HTTP Server.



**Figure 2 HTTP traffic pattern**

Packet session represents that web page is being transferred from HTTP Server to HTTP Client, and the reading interval represents the time that Client spending reading the webpage. The amount of information transferred from the Server to Client during the packet session is governed by the web page structure. A webpage is usually composed of a main object and several embedded objects. The total amount of traffic transferred from HTTP Server to HTTP Client is equivalent to the size of the main object and a number of the embedded objects, where the model parameters of HTTP Server traffic are specified in Table 2.

The HTTP Client sends an HTTP request packet, which has a constant size of 350 bytes in order to initiate an HTTP Server traffic event.

From the statistics presented in the literature, a 50%-50% distribution of HTTP versions between HTTP 1.0 and HTTP 1.1 has been found to closely approximate web browsing traffic in the internet. Studies also show that the maximum transmit unit (MTU) sizes most common to the internet are 576 bytes and 1500 bytes (including the TCP header) with a distribution of 24% and 76% respectively.

Thus, the HTTP Server web traffic generation is that, at first, to create an HTML page using the HTML page statistics, with either MTU size is 1500 bytes or 576bytes, and then to download the main and the embedded objects using either HTTP/1.0-burst transport or HTTP/1.1- persistent transport.

**Table 2: Parameters for HTTP traffic model**

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Distribution | Parameters | PDF |
| File  size (S) | Truncated Logonormal | Mean = 10710 bytes  SD = 25032 bytes  Min = 100 bytes  Max = 2 Mbytes  (before truncation) | if x > max or x < min, discard and generate a new value for x |
| Embedded  Object size (SE) | Truncated Logonormal | Mean = 7758 bytes  SD = 126168 bytes  Min = 50 bytes  Max = 2 Mbytes  (before truncation) | if x > max or x < min, discard and generate a new value for x |
| Number of  embedded  objects per  page (Nd) | Truncated  Pareto | Mean = 5.64  Max. = 53  (before truncation) | Subtract k from the generated random  value to obtain Nd  if x > max, discard and regenerate a new value for x |
| Reading  time (Dpc) | Exponential | Mean = 30 sec |  |
| Parsing  time (Tp) | Exponential | Mean = 0.13 sec |  |

In addition to the HTTP request packet from HTTP Client to HTTP Server, the reciprocal TCP Ack to the HTTP traffic, from HTTP Client to HTTP Server is generated as follows:

Step 1: 40 Byte TCP Ack, inter-arrival interval is equal to 1ms delay from HTTP traffic frame reception at the HTTP Client

Traffic Model Class Identifier for HTTP is HTTP

**Evaluation Metrics**

* MAC Throughput, latency

## References for traffic models

1. **11-13-1334-05-video traffic modeling**
2. **11-13-1335-04- video-traffic-modeling-word with details**
3. **11-13-1162-01-hew-vide-categories-and-characteristics**
4. **11-13-1059-01-hew-video-performance-requirements-and-simulation-parameters**
5. **11-09-0296-16-00ad-evaluation-methodology.doc**
6. **Rongduo Liu et al., “An Emperical Traffic Model of M2M Mobile Streaming Services ”, International conference C on Multimedia information networking and security, 2012**
7. **JO. Rose, “ Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems ”, Tech report, Institute of CS in University of Wurzburg**
8. **Savery Tanwir., “A survey of VBR traffic models”, IEEE communication surveys and tutorials, Jan 2013**
9. **Aggelos Lazaris et al., “A new model for video traffic originating from multiplexed MPEG-4 videoconferencing streams”, International journal on performance evaluation, 2007**
10. **A. Golaup et al., “Modeling of MPEG4 traffic at GOP level using autoregressive process”, IEEE VTC, 2002**
11. **K. Park et al., “Self-Similar network traffic and performance evaluation”, John Wiley&Son, 2000**
12. **M Dai et al., “A unified traffic model for MPEG-4 and H.264 video traces”, IEEE Trans. on multimedia, issue 5 2009.**
13. **L Rezo-Domninggues et al., “Jitter in IP network: A cauchy approach”, IEEE Comm. Letter, Feb 2010**
14. **Hongli Zhang et al., “Modeling Internet link delay based on measurement”, International conference on electronic computer technology, 2009.**
15. **IEEE 802.16m-08/004r5, IEEE 802.16m Evaluation Methodology Document (EMD)**
16. **Yingpei Lin et al., 11-13-1133-00-0hew-virtual-desktop-infrastructure-vdi**
17. **Yingpei Lin et al., 11-13-1438-00-0hew-traffic-observation-and-study-on-virtual-desktop-infrastructure**
18. **Yingpei Lin et al., 11-14-0056-01-0hew-traffic-model-on-virtual-desktop-infrastructure**
19. **Yingpei Lin et al., 11-14-0594-01-00ax-insert-virtual-desktop-infrastructure-vdi-traffic-model-content-for-hew-simulation-scenarios**
20. **11-13/486, “HEW video traffic modeling” Guoqing Li et al, (Intel)**

# Appendix 3 – RBIR and AWGN PER Tables

The embedded spreadsheet contains the RBIR and AWGN PER tables. The results under the Mean column shall be used for all simulations and calibration.



# Appendix 4 – Packet Reception and Preamble Detection Procedure

The original discussion of the following text can be found in [25].

1. PPDU capture: if more than one PPDU arrives at a receiver within a PPDU capture window of A nsec, then the strongest of the arriving PPDUs shall be the one captured (A is TBD but with one option of 800ns).
   1. The PPDU capture window of A nsec starts at the first arrival PPDU with rx power higher than rx sensitivity;
   2. A PPDU with rx power lower than rx sensitivity is dropped, which does not impact current receiver status;
2. At the end of the PPDU capture window, the receiver locks to the strongest PPDU whose preamble is to be decoded and other PPDUs coming within the duration of the preamble of the strongest PPDU are considered as interference and shall be not detected.
   * Take the whole preamble of each PHY PPDU as a standalone sub-frame.
     + The packet length in bytes used in preamble PER computation is calculated based on the assumption of 3-byte/4us (MCS0)
   * Model the preamble decoding the same as a sub-frame decoding
   1. If preamble passes (i.e., successfully decoded), the receiver continues to receive the rest part of the PPDU, i.e., to decode each MPDU;
      * If successfully decoding of a MAC frame, defer for NAV;

* Control frame is dealt with as a standalone sub-frame
* Apply NAV cancellation for RTS according to current std spec
  + - Otherwise, set CCA to busy for the entire PPDU duration if rx power higher than TBD [rx sensitivity or CCA-SD].
    - At the same time just after the end of the preamble, a preemption window (the window size is TBD in [0, PPDU duration]) begins during which time if a new PPDU arrives with rx power at least PdB above the current reception, then the current reception is terminated and the PPDU capture phase is re-entered with this new PPDU (P is expected to be have a value of about 9 dB)
  1. If preamble fails, the receiver terminates current reception
     + The entire PPDU fails
     + The receiver is unlocked again and then CCA-ED threshold is used to determine if the medium is busy.