Abstract—To evaluate the performance of distributed medium access control (MAC) layer of emerging ultra wideband (UWB) and 60-GHz millimeter wave (mmWave) wireless personal area networks (WPANs) based on distributed reservation protocol (DRP) of ECMA-368 and ECMA-387 standards, we provide an analytical model for the probability distribution of steady state throughput as a function of the network dimensions size, number of devices, antenna beamwidth, transmission range, bit error rate (BER), antenna disturbance rate (ADR), aggregation length, payload size, and incoming frame rate. Using this model, the optimal payload size is computed as an example of optimization problem. We also propose a formula for the starvation and fairness and using them, we found a major flaw in DRP, i.e., its poor fairness. We develop a fair DRP (FDRP) protocol and modify our model to cover it. The models are evaluated by simulation of ad-hoc network scenarios and results show that the average error is less than 2 percent.

Index Terms—Throughput, medium access control, ultra wideband, 60-GHz mmWave, wireless personal area network

1 INTRODUCTION

Wireless personal area networks play a major role in file transfer, video and sound transmission, short-range networks, and cable removal between equipments and peripheral devices. The dominant technologies in future WPANs will be UWB and 60 GHz mmWave [1]. UWB is defined in the 3.1-10.6 GHz band and can provide data rates up to 480 Mbps. This band is used by licensed technologies and hence, a rigorous restriction is applied by federal communications commission (FCC) to power level of compliant devices. The 60-GHz technology, on the other hand, is defined in a 9 GHz unlicensed band around the 60 GHz frequency and can provide data rates up to 25 Gbps. After failure of IEEE 802.15.3a in providing a standard for UWB WPAN, the multiband orthogonal frequency division multiplexing UWB (MB-OFDM UWB) is promoted by ECMA-368 standard [3] which unlike IEEE 802.15.3a defines an ad-hoc network without central coordinated access to the wireless medium. This distributed architecture led to the success and global acceptance of this standard for UWB WPAN [4].

Three standards are defined for the 60 GHz WPANs, namely, WirelessHD [5], IEEE 802.15.3c [6], and ECMA-387 [2]. While the first two standards define centralized MAC architectures, the last one defines an unstructured MAC mechanism similar to ECMA-368. Because of success achieved by ECMA-368, it seems in future, ECMA-387 standard will occupy a considerable portion of the 60 GHz WPANs’ market share, similar to UWB technology.

Analysis and modeling of the MAC layer can help the standard bodies to improve the performance of protocols. The throughput model can also be implemented and used in the MAC layer to control incoming traffic and quality of service (QoS) by informing the higher layers. Also, vendors can compare different standards and choose one with best performance. While most scientific works reported in the literature model the performance of IEEE 802.15.3a [7], [8] and IEEE 802.15.3c [9], [10], [11], some few authors modeled the MAC layer of ECMA-368.

While some of these works model the prioritized contention access (PCA) protocol [12], [13], some few papers modeled some features of DRP [14], [15], [16] among them Liu et al. [16] proposed a model on throughput performance. They used queueing theory principles and ignored competition between devices. The effect of directional antenna, network dimensions size, number of devices, transmission range, and aggregation length is not included in their model.

DRP has two key features: its Beaconing procedure and its distributed reservation. In our previous work, we modeled its Beaconing protocol [17]. Here we focus on the second aspect of standard and propose a model on throughput of DRP based on the number of devices in network, transmission range (Tx range), antenna beamwidth, network dimensions size, ADR, BER, traffic load, payload size, and aggregation size. Use of steerable directional antennas in ECMA387 is another key feature and its effect on throughput is covered in the developed model. We also explain the weak fairness in the DRP and propose a fair DRP (FDRP). We show by modeling and extensive simulations how it improves the fairness.

The remainder of this paper is organized as follows: In Section 2, we briefly discuss the DRP protocol defined in ECMA-368 and 387 standards. Section 3 presents the proposed throughput model. An optimization problem is presented and solved in Section 4 which finds the optimum payload size. FDRP is explained and modeled in Section 5. Correctness of models is evaluated in Section 6. Finally, Section 7 concludes the paper.

Notations. Notations of variables used in the paper are presented in Table 1.
TABLE 1
Notations of Variables and Parameters in The Paper

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>ADR, antenna disturbance rate (times per second)</td>
</tr>
<tr>
<td>B</td>
<td>BPL of the device</td>
</tr>
<tr>
<td>F</td>
<td>Fairness</td>
</tr>
<tr>
<td>f_{in}</td>
<td>Number of incoming frames in duration of a SF</td>
</tr>
<tr>
<td>f_{safe}(Z)</td>
<td>Number of frames sent safely with fixed antenna</td>
</tr>
<tr>
<td>f_{raw}(Z)</td>
<td>Number of frames sent out successfully in Z MASs</td>
</tr>
<tr>
<td>F_X(\cdot)</td>
<td>CDF of random variable X</td>
</tr>
<tr>
<td>f(Z)</td>
<td>Number of frames transferrable in Z MASs</td>
</tr>
<tr>
<td>L</td>
<td>Number of MAS per SF (256)</td>
</tr>
<tr>
<td>M</td>
<td>Maximum allowable BPL (mMaxBPLength)</td>
</tr>
<tr>
<td>N</td>
<td>Number of devices in the network</td>
</tr>
<tr>
<td>N_1</td>
<td>Number of first order neighbors of a device</td>
</tr>
<tr>
<td>N_2</td>
<td>Number of second order neighbors of a device</td>
</tr>
<tr>
<td>n_{ACK}</td>
<td>ACK payload size in octets</td>
</tr>
<tr>
<td>n_{agg}</td>
<td>Number of frames aggregated in one aggregation</td>
</tr>
<tr>
<td>n_{AMH}</td>
<td>MAC header size of the ACK frame (10 octets)</td>
</tr>
<tr>
<td>n_{ATIF}</td>
<td>Antenna training indicator field size (3 octets)</td>
</tr>
<tr>
<td>n_{ATS}</td>
<td>Antenna training sequence size (32768 bits)</td>
</tr>
<tr>
<td>n_{FPH}</td>
<td>Fixed length PHY header size (15 octets)</td>
</tr>
<tr>
<td>n_{FSS}</td>
<td>Frame synchronization sequence size (2048 bits)</td>
</tr>
<tr>
<td>n_H</td>
<td>Number of octets in the header of aggregated frame</td>
</tr>
<tr>
<td>n_HACK</td>
<td>Number of ACK header octets</td>
</tr>
<tr>
<td>n_HCS</td>
<td>Size of header check sequence (2 octets)</td>
</tr>
<tr>
<td>n_P</td>
<td>Payload size (Octets)</td>
</tr>
<tr>
<td>n_T</td>
<td>Payload size after RS encoding</td>
</tr>
<tr>
<td>n_RSP</td>
<td>Size of RS parity (16 octets)</td>
</tr>
<tr>
<td>n_seq</td>
<td>Number of segments in segmentation of each frame</td>
</tr>
<tr>
<td>n_{VPH}</td>
<td>Variable length PHY header size (4n_{seq} octets)</td>
</tr>
<tr>
<td>F_X(\cdot)</td>
<td>PMF of random variable X</td>
</tr>
<tr>
<td>r</td>
<td>Tx range</td>
</tr>
<tr>
<td>S</td>
<td>Starvation</td>
</tr>
<tr>
<td>T</td>
<td>Throughput (bits per second)</td>
</tr>
<tr>
<td>T_{ACK}</td>
<td>Time duration of sending an ACK frame</td>
</tr>
<tr>
<td>T_{agg}</td>
<td>Time duration to send an aggregated frame</td>
</tr>
<tr>
<td>T_b</td>
<td>Time duration of sending one bit (0.5787 ns)</td>
</tr>
<tr>
<td>T_G</td>
<td>Guard time (4.7 ms)</td>
</tr>
<tr>
<td>T_{MAS}</td>
<td>Duration of one MAS (64 μs)</td>
</tr>
<tr>
<td>T_{SIFS}</td>
<td>Short interframe spacing (2666 ns)</td>
</tr>
<tr>
<td>T_S</td>
<td>Duration of a SF (= L × T_{MAS})</td>
</tr>
<tr>
<td>T_x</td>
<td>Time duration of an aggregated frame and its ACK</td>
</tr>
<tr>
<td>Z</td>
<td>Number of reserved MASs by a device</td>
</tr>
<tr>
<td>Z_{1st}</td>
<td>The first reserved MAS in a block of 16 MASs</td>
</tr>
<tr>
<td>Z_{req}</td>
<td>Z in the case when relinquish request is allowed</td>
</tr>
<tr>
<td>α</td>
<td>Number of available MASs for a device</td>
</tr>
<tr>
<td>η_X</td>
<td>Mean of random variable X</td>
</tr>
<tr>
<td>θ_1,θ_2,θ_3</td>
<td>Intermediate parameters used for simplicity</td>
</tr>
<tr>
<td>λ</td>
<td>Traffic load of the device in frames per second</td>
</tr>
<tr>
<td>ρ</td>
<td>Bit error rate</td>
</tr>
<tr>
<td>σ_X</td>
<td>Variance of random variable X</td>
</tr>
<tr>
<td>τ</td>
<td>Approximate throughput computed by a device</td>
</tr>
<tr>
<td>φ_{dB}</td>
<td>3 dB antenna beamwidth of devices</td>
</tr>
</tbody>
</table>

2 DRP

In this section, we briefly discuss the DRP protocol. We base our discussion on ECMA-387. In ECMA-387, three types of devices are defined: Type A uses steerable directional antenna while type B and C use omni-directional antennas. Devices of type A and B use DRP protocol while devices of type C use a simple master-slave protocol. In this paper, we assume all devices are of type A or B (in ECMA-387) or UWB-based (in ECMA-368). In the next section, we explain the beacon protocol of ECMA-387 standard.

2.1 Beacon Protocol

In both ECMA-368 and ECMA-387, time is divided into equal durations known as SF. A SF itself is divided into 256 equal size units called MAS. The devices align their SF with the slowest device in the network.

Each device randomly selects one of the first available MASs (or Beacon slots as shown in Fig. 1) in the SF for sending a frame called beacon which is used for synchronization and other control operations with other devices. A device considers a beacon slot as unavailable if one of its neighbors communicate in it. A device is considered as neighbor by any other device, if its beacon can be received directly. The highest number beacon slot up to which a device can select for beaconing is called mMaxBPLength. The remaining MASs after the last beacon slot are used for data communication. This last beacon slot indicates beacon period length (BPL) which is variable from device to device. Each device provides a beacon period occupancy information element (BPOIE) in which beacon slots adopted by neighbor devices are indicated and announces it using its beacon. Fig. 1 shows an example of a SF structure in which beacon slots and beacon period are depicted. The beacon protocol has more details which are skipped here. In the next section, we briefly explain the data communication of DRP.

2.2 Data Communication

DRP is a kind of distributed time division multiple access (TDMA). In this protocol, each device in its beacon announces the MASs it intends to reserve for sending its frames. These MASs should be available for the device and its target. To do this, each device holds a DRP information element (IE) and announces it through its beacon. For example, suppose device X wants to send some data frames to device Y. To request a MAS for reservation, X sets parameters Owner and Target of its intended MASs to its own and Y’s ID, respectively, and Reservation Status to available in its beacon. If Y agrees to receive from X, it confirms the request in its next beacon by setting parameters Owner and Target of the requested MASs to X and its own ID, respectively, and Reservation Status to available. Upon receiving this beacon, X sets Reservation Status of the confirmed

2 In ECMA-387, mMaxBPLength = 48.
MASs to reserved. Finally, Y sets the Reservation Status of the requested MASs to reserved and reservation is done. Thereafter, X starts sending its data frames in the reserved MASs.

The other devices in the vicinity consider those MASs as occupied (unavailable) and do not try to reserve them. X sends its frames only in the reserved MASs until any time it wants. Both X and Y can defer from this reservation by simply setting Reservation Status of the reserved MASs to available and Owner and Target, appropriately, to release the MASs. After releasing a MAS, other devices can try to reserve them. The device which releases a MAS, can reserve it again by doing the reservation process from scratch.

In this paper, we consider B-ACK, in which multiple frames are aggregated in a block and sent together. An ACK frame is sent back in response to them in which the acknowledgements of all aggregated frames are included.

The standard uses a backoff mechanism for the cases when two devices try to reserve a MAS simultaneously. We believe that such contention conditions occur very rarely, and so has no significant effect in the throughput of the whole system. So, we ignore its effect in our model. The results given in Section 6 support our claim.

To request a device to release one or more MASs from existing reservations, the relinquish request (RR) is defined. For this purpose, each device announces via its beacon for which MASs the other devices can request for relinquish. However, the maximum number of MASs which can be hold (not announced) are limited. When a device needs more MASs than the available ones, it requests for relinquish of some unsafe MASs (relinquish eligible MASs) by indicating a device which is the owner of those MASs. The owner shall release the unsafe MASs requested for relinquish upon receiving this beacon. When a device reserves some MASs after requesting for relinquish, it is allowed to hold them for 32 SFs. After which it should announce unsafe MASs in the same way.

2.3 Antenna Training and Tracking Protocol (ATTP)
To achieve higher antenna gains, an antenna training and tracking protocol is defined by which devices can adjust their antenna weights. ATTP is based on an RTT/CTT (request to train/clear to train) mechanism. With this protocol, the device transmits some pre-specified bitstream in different directions and the receiver sends the antenna weights back. This procedure is also repeated in reverse direction.

The antenna direction may change during communication because of device movements or changes in the channel. A device can use the physical (PHY) layer parameters and also the BER level to decide whether to retrain the antenna parameters. If so, the devices starts ATTP.

Here, we provided a brief description of the main procedures related to our work. For more details, the reader may refer to [2]. In the next section, we develop our model on throughput.

3 Modeling the Throughput
In this section, we develop an analytical model for computation of the probability distribution of throughput. We begin our analysis by the following assumptions:

1) N devices of the same type are scattered randomly in unit square with uniform distribution.
2) The Tx range and 3 dB antenna array beamwidth for all devices are \( r \) and \( \phi_{\text{stdB}} \), respectively.
3) Each device can only communicate in its \( \phi_{\text{stdB}} \) beam up to range \( r \) (flat antenna model).
4) We assume that the network is in steady state condition, i.e., all devices synchronize their SFs and they can retrain their antennas before \( \text{MaxLostBeacons} \) in small movements and channel changes, so the network is considered to be stable. Moreover, the movements are considered small such that the devices do not exit from each others’ neighborhood.
5) We also suppose that all devices completed BP contraction procedure. However, the problem can easily be changed to the case with no BP contraction by changing the BPL [17].
6) Devices have incoming frame rate of \( \lambda \) frames per second (fps) with Poisson distribution.
7) The BER for payload of frames is \( p \).
8) Some events cause the antenna direction of devices to exit from true direction. We call them antenna disturbance event (ADE). These events follow Poisson distribution with parameter \( a \) events per second. \( a \) is called antenna disturbance rate (ADR) in this paper.
9) The incoming frames have payloads of constant size \( n_{\text{pl}} \).
10) We assume that devices use the same channel.

Remark 1. Note that with \( \phi_{\text{stdB}} = 2\pi \), the problem reduces to omni-directional case corresponding to ECMA-368 and devices of type B in ECMA-387.

Remark 2. We introduced almost all affecting parameters from MAC’s point of view to develop a complete model. By the way, the reader can omit any parameter by setting appropriate values. For example, by setting \( a = 0 \), the effect of ADE can be eliminated from the proposed model.

Remark 3. We assume that all devices experience the same amount of BER. This is reasonable because all devices are in similar channel conditions. However, one can consider a probability distribution for BER and modify our model accordingly.

We take three steps to develop a formula for the probability distribution of throughput:

1) We first relate the throughput to the number of reserved MASs by a device (Section 3.1).
2) The data frame structure and timing parameters is then addressed in some formulas (Section 3.2).
3) Finally, the probability distributions are computed (Section 3.3).

3.1 Throughput
We use the following formula for computing the throughput:

\[
T = \frac{8n_{\text{pl}} f_{\text{out}}(Z)}{T_{\text{SF}}}
\]

in which \( n_{\text{pl}} \), \( T_{\text{SF}} \) and \( f_{\text{out}}(Z) \) stand for the payload size of frames in octet, duration of a SF, and the number of frames
transmitted out of device successfully in \( Z \) reserved MASs in one SF. The number 8 is entered to convert from byte to bit, thus \( T \) is in units of bps.

In our model, we differentiate between errors caused by antenna mismatch of the communicating devices and errors caused by wireless channel imperfections. While the latter affects the data frames continually in all SFs, the first one occurs randomly (and hence modeled as a Poisson process) and when it occurs, in the next few SFs all transmitted data frames will be corrupted. In our model, to separate these events, we define three parameters: \( f_{\text{out}}(Z) \), \( f_s(Z) \), and \( f(Z) \), the number of frames which can be transmitted in \( Z \) MASs, considering both types of events, considering only wireless channel imperfections, and considering ideal channel, respectively. In other words, from \( f(Z) \) we can compute \( f_s(Z) \) by adding the effect of antenna mismatch events and from \( f_s(Z) \) we can obtain \( f_{\text{out}}(Z) \) by applying the effect of wireless channel imperfections:

\[
f_{\text{out}}(Z) = f_s(Z)(1 - \rho)^{8(n + n_{\text{fl}})}.
\]

Here, for the sake of simplicity, we assume that headers are received without error and the only sources of error are frame payloads and their ACKs (the term \( 6 + n_{\text{fl}} \)). For each data frame, 6 bytes are included in its related ACK. We neglect the header errors, since the headers are coded with more robust coding. However, one can consider another error rate for header and multiply Eq. (2) by another similar term. In our simulations, we considered a BER much less than payload BER for header part.

In Eq. (2), we need \( f_s \). To compute it, we proceed as follows: we suppose that when an ADE occurs in a SF, all received and sent frames after that event shall be destroyed. However, in our simulations, we magnify BER in such conditions. We also assume that the device senses the ADE at the end of SF in which the ADE occurred. In the next SF, it requests for antenna training and finally, in the following SF, training takes place. The data communication restarts at the next SF.

Now, we relate \( f_s(Z) \) to \( f(Z) \). Suppose that the reservation block starts at \( k \)th MAS. As we assumed:

1) If in two previous SFs at least one ADE occurs, \( f_s(Z) \) would be zero.
2) If at least an ADE occurs in the time duration between the start of current SF to the end of \( k \)-1st MAS, \( f_s(Z) \) would be zero.
3) If in the duration \( \left[ \frac{s-1}{f(Z)}, \frac{s}{f(Z)} \right] \), at least one ADE occurs and no other ADE occurred before that, \( f_s(Z) = g - 1 \), where \( g \) is a natural number.
4) Otherwise, it would be \( f(Z) \).

This leads us to the following formula:

\[
f_s(Z) = \sum_{g=1}^{f(Z)} (g - 1)e^{-2aT_{\text{mas}}(2L + k - 1 + g - 1)}
\times (1 - e^{-2aZ/f(Z)})
+ f(Z)e^{-2aT_{\text{mas}}2L + k - 1 + Z}
\]

in which, \( L = 256 \) is the number of MASs in each SF. The summation corresponds to the third condition, and \( g - 1 \) is the number of frames. The first exponential is the probability that no ADE occurs before duration \( \frac{s-1}{f(Z)} \), and the last term in the summation, \( (1 - e^{-2aZ/f(Z)}) \), denotes the probability that at least one ADE occur in that duration, given that ADEs follow Poisson distribution. The second term in Eq. (3), corresponds to the fourth argument, and the exponential term is the probability that before the end of reservation block, no ADE occurs.

In Eq. (3), the minimum and maximum value for \( k \) are \( B + 1 \) and \( L - Z + 1 \), respectively. So, as an approximation, we use the average value for \( k \) to be

\[
k = \left\lceil \frac{L + B - Z}{2} \right\rceil,
\]

where, \( \lceil . \rceil \) denotes ceiling operator.

Now, we would like to find the function \( f(Z) \), i.e., given a number of MASs, how many frames can be transmitted? For the sake of simplicity, we suppose that all MASs are reserved consecutively. To answer this question, consider that the available time is \( Z \times T_{\text{mas}} - T_G \), where \( T_{\text{mas}} \) and \( T_G \) denote the MAS duration and guard time, respectively. In this duration, we first send as much aggregated frames of size \( N_{\text{agg}} \) incoming frames as we can (the first term of Eq. (5)). Then, we send an aggregated frame with smaller size of \( n_{\text{agg}} \) frames in remaining time duration \( (T_{\text{rem}}) \) of reserved block (the second term):

\[
f(Z) = \left\lfloor \frac{Z \times T_{\text{mas}} - T_G}{T_{\text{tx}}(N_{\text{agg}})} \right\rfloor N_{\text{agg}}
+ \max(n_{\text{agg}}T_{\text{tx}}(N_{\text{agg}}) \leq T_{\text{rem}}),
\]

where \( \lfloor . \rfloor \) denotes flooring operator. In the first term of Eq. (5), we multiply the number of aggregations of size \( N_{\text{agg}} \) by \( N_{\text{agg}} \) to obtain the number of frames which are aggregated in aggregations of size \( N_{\text{agg}} \). Also, \( T_{\text{tx}}(n_{\text{agg}}) \) is the time duration needed to transmit \( n_{\text{agg}} \) aggregated frames and receiving their ACK. This is dealt with in Eq. (11). Since each SF consists of \( L \) MASs, \( T_{\text{mas}} = T_{SF}/L \). \( N_{\text{agg}} \) is an adjustable parameter in the MAC layer which can be set from 1 up to 64 frames. \( T_{\text{rem}} \) can be computed as:

\[
T_{\text{rem}} = Z \times T_{\text{mas}} - T_G - \left\lfloor \frac{Z \times T_{\text{mas}} - T_G}{T_{\text{tx}}(N_{\text{agg}})} \right\rfloor T_{\text{tx}}(N_{\text{agg}}).
\]

Suppose that a device observes \( a \) available MASs in a SF, the number of MASs it reserve would be:

\[
Z(a) = \begin{cases} 
  a, & f_{\text{in}} + f(Z) - f_{\text{out}}(Z) \geq f_{\text{out}}(Z) \\
  f^{-1}(f_{\text{in}} + f(Z) - f_{\text{out}}(Z)), & \text{o.w.} 
\end{cases}
\]

in which, \( f^{-1} \) denotes the inverse function of \( f \) and it returns the minimum number of required MASs sufficient to send the number of frames given as its argument. Eq. (7) states that when the average number of incoming frames is more than the average number of outgoing frames, in steady state, all available MASs will be reserved. Therefore, \( Z(a) = a \). Otherwise, the number of reserved MASs will be minimum number of MASs needed to send \( f_{\text{in}} = LT_{\text{SF}} \) incoming frames and \( f(Z) - f_{\text{out}}(Z) \) frames returning to queue because of bit errors occurred in them.
Algorithm 1. Computation of $Z(\alpha)$

Input: Scalars $\alpha, n_{PL}, f_{in}, \rho, \alpha, B$ and the vector $f$ of length $L - 1$

Output: The scalar $Z$

1. $n_{MAS} \leftarrow \text{needemas}(f_{in}, f)$
2. if $\text{isempty}(n_{MAS})$ or $n_{MAS} > \alpha$ then $Z \leftarrow \alpha$
3. else
4. $n_{MAS_{new}} \leftarrow \text{needemas}(f_{in} + f[n_{MAS}])$
5. $f[n_{MAS}](1 - \rho)^{(n_{MAS} + n_{PL})}, f)$;
6. if $\text{isempty}(n_{MAS_{new}})$ or $n_{MAS_{new}} > \alpha$ then $Z \leftarrow \alpha$
7. while $n_{MAS_{new}} > n_{MAS}$ and $n_{MAS_{new}} < \alpha$ do
8. $n_{MAS} \leftarrow n_{MAS_{new}}$;
9. compute $f_s$ using $Z = n_{MAS}$ from Eq. (3);
10. $n_{MAS_{new}} \leftarrow \text{needemas}(f_{in} + f[n_{MAS}])$
11. $f[n_{MAS}](1 - \rho)^{(n_{MAS} + n_{PL})}, f)$;
12. if $\text{isempty}(n_{MAS_{new}})$ or $n_{MAS_{new}} > \alpha$ then $Z \leftarrow \alpha$;
13. break;
14. end
15. end
16. $Z \leftarrow n_{MAS_{new}}$;
17. end
18. end

Since in Eq. (7), the term $Z$ appears in the right side, we propose the iterative Algorithm 1 to compute it. In this algorithm, $\text{needemas}(f_{in}, f)$ is a function searching $f(Z)$ for the first $Z$ for which $f_{in} < f(Z)$. More precisely, $f(Z)$ is a vector of size $L$ for different values of $Z \in \{1 \ldots L\}$, where each value corresponds to the number of frames which can be transmitted in $Z$ reserved frames. In other words, needemas is the inverse function of $f(Z)$, i.e., needemas $(f_{in}, f) = f^{-1}(f_{in})$. The first two lines of algorithm check whether the number of required MASs to fit $f_{in}$ frames is more than $\alpha$ and if so, it sets $Z = \alpha$. Analogously, in fourth and fifth lines, it checks whether the number of required MASs to fit $f_{in} + f_{out}$ frames is more than $\alpha$ and if so, it sets $Z = \alpha$. Otherwise, in the while loop, it computes $f_s(Z)$ for $Z = n_{MAS}$ and checks whether the number of required MASs to fit $f_{in} + f_{out}$ frames is more than $\alpha$ and if so, it sets $Z = \alpha$ and exits. The while loop proceeds until the value of $n_{MAS}$ does not change or becomes greater than $\alpha$.

To find the number of available MASs in each SF, $\alpha$, it is sufficient to subtract the BPL $B$ and the number of reserved MASs by previous devices from $L$:

$$\alpha(d, N_2) = \max \left\{ L - B - \sum_{i=1}^{d-1} Z(\alpha(i, N_2)), 0 \right\}$$

in which, $N_2$ denotes second order neighbors (SON) and $d$ is used to state that among $N_2 + 1$ devices (the device itself and $N_2$ SONs), the device was the $d$th one which tried to take part in reservation. The concept of SON introduced in [17] means a device that is whether the neighbor or neighbor of a neighbor of a given device.

As the final point in this section, we explain how to apply the effect of relinquish request to our model. Based on standard, in each block of 16 consecutive MASs, a device can announce at most $\sigma$ MASs as $safe$, i.e., it can hold at most $\sigma$ MASs and does not allow other devices to request for relinquish of them. The rule used by standard is as follows:

$$\sigma = \begin{cases} \frac{2}{3}Z^{1st}, & Z^{1st} \in \{0 \ldots 4\} \\ 4, & Z^{1st} \in \{5 \ldots 12\} \\ 16 - Z^{1st}, & Z^{1st} \in \{13 \ldots 15\} \end{cases}$$

where the superscript $\text{rel}$ in $Z_{rel}$ is used to emphasis on using relinquish request. The fourth row in Eq. (10) is computed from the second row of Eq. (7). The other formulas of this section remain the same.

In above formulas, $T_{T_{rel}}$ is a constant and should be computed from the standard framing structure and timing parameters. This is the subject of the next section.

### 3.2 Timing Parameters of Frames

Fig. 2 shows the timings defined in ECMA-387 for B-ACK method. Each aggregated frame is followed by a waiting time of short interframe spacing (SIFS) and then by an ACK and another SIFS. A guard time is also considered at the end of reservation block to prevent collision between adjacent reservation blocks. Therefore, the time duration to send an aggregated frame and receiving its ACK is:

$$T_{T_{rel}} = T_{agg} + T_{ACK} + 2T_{SIFS}.$$ (11)

Based on the frame structure of ECMA-387, the time duration to send an aggregated frame can be obtained from:

$$T_{agg} = 8T_b(n_H + n_{PL} + n_{FSS} + n_{ATS}),$$ (12)

where $T_b$ is the time duration of sending one bit. In Eq. (12), $n_H$ denotes the number of octets in the header of aggregated frame:

$$n_H = \left\lfloor \frac{4n_{agg} + C_1}{4n_{agg} + C_2} (4n_{agg} + C_2 + 16) \right\rfloor$$

3. In mode A2 of ECMA-387, $T_b = 0.5787$ nsec.
in which
\[ C_1 = n_{FPH} + n_{VPH} + n_{ATIF} + 10 + n_{HCS} + n_{RSP}, \]
\[ C_2 = 30 + 4n_{seg}. \]  
(14)

In Eq. (13), \( n_{agg} \) is the number of frames aggregated in one aggregation and \( \lceil . \rceil \) denotes ceiling operator. Also, in Eq. (14), \( n_{seg} \) denotes the number of segments in segmentation of each frame. In our simulations and models, it is set to 1.

In Eq. (12), \( n_{pl}' \) indicates the number of payload octets after Reed-Solomon (RS) encoding. Based on the frame structure of ECMA-387, it can be obtained from:
\[ n_{pl}' = 240 \times \left\lceil \frac{4n_{agg}[n_{PL}/4]}{224} \right\rceil, \]
(15)
where \( n_{PL} \) is the number of payload octets given to the MAC layer.

Similarly, we compute the same values for ACK frame. Number of ACK header octets is:
\[ n_{HACK} = \left\lceil \frac{C_3}{C_2} \right\rceil (C_2 + 16) \]  
(16)
in which
\[ C_3 = n_{FPH} + n_{VPH} + n_{ATIF} + n_{AMH} + n_{HCS} + n_{RSP} \]
(17)
and the number of ACK payload octets is:
\[ n_{ACK} = 240 \times \left\lceil \frac{6 + n_{agg}}{224} \right\rceil. \]  
(18)

Finally, sending an ACK frame takes
\[ T_{ACK} = 8T_b(n_{HACK} + n_{ACK} + n_{FSS} + n_{ATS}). \]  
(19)

Equations (11)-(19) are derived based on the framing commands given in ECMA-387.

So far, we related the throughput to some constants and three random variables (RV), i.e., \( B, d \) and \( N_z \). Therefore, the throughput is also a RV. The distribution functions of \( B, d \) and \( N_z \) are computed in [17] and their final results are summarized in appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2015.2417881. In the next section, we find the distribution function of throughput.

### 3.3 Probability Distribution of Throughput

In Eq. (1), we observed that \( T \) is a function of RV \( f_s \). Hence,
\[ P_T(T) = \sum_{\{f_s|T=T(f_s)\}} P_{f_s}(f_s). \]  
(20)

On the other hand, from Eq. (3), \( f_s \) is a function of \( Z \), and it takes values \( f_s(Z) \) with probabilities \( P_Z(Z) \). Therefore,
\[ P_{f_s}(f_s) = \sum_{\{Z|f_s=f_s(Z)\}} P_Z(Z). \]  
(21)

To compute \( P_Z(Z) \), we use the total probability theorem as follows:
\[ P_Z(Z) = \sum_{N_z=1}^{N_z+1} \sum_{d=1}^{N_z} P_Z(Z|d,N_z) P_d(d|N_z) P_{N_z}(N_z). \]  
(22)

Similarly, from Eq. (7), \( Z \) is a function of RV \( \alpha \) which takes values of \( Z(\alpha) \) with probability \( P_\alpha(\alpha) \). Therefore,
\[ P_Z(z|d,N_z) = \sum_{\{\alpha|Z(\alpha)\}} P_\alpha(\alpha|d,N_z). \]  
(23)

Finally, from Eq. (8) \( \alpha \) is a function of \( B, d \) and \( N_z \). Therefore,
\[ P_\alpha(\alpha|d,N_z) = \left\{ \begin{array}{ll}
P_{\alpha}(L - \alpha), & d = 1 \smallbreak 
\sum_{b=2}^{M} P_{\alpha}(b) P_{\alpha}(L - b - \alpha|d = 1), & d = 2 \\
\frac{1}{\sqrt{2\pi\sigma_{\alpha}^2}} e^{-\frac{1}{2}(x-\alpha)^2}, & \text{o.w.,}
\end{array} \right. \]  
(24)
where, \( P_\alpha(.) \) denotes the PMF of BPL and \( P_\alpha(\alpha|d = 1) \) is the PMF of the number of reserved slots by the first device. \( M = \max \text{BPLength} \). The first row holds because when the device is the first one which tries to reserve MASs, it observes \( L - B \) available MASs, and hence \( P_\alpha(\alpha) = P_{\alpha}(L - \alpha) \). Similarly, the second row holds because when the device is the second one which tries to reserve MASs, it observes \( L - B - Z_1 \) available MASs, where \( Z_1 \) denotes the number of reserved MASs by first device. \( B \) takes values of \( 2, \ldots, M \), and the summation in second row would be \( P_\alpha(\alpha|d = 2) \). In the same manner, we can proceed for larger values of \( d \), but it requires a huge number of computations. So, in the third row we used central limit theorem which states that the sum of \( \alpha \) independent RVs tends to normal distribution with mean and variance equal to sum of means and variances of each RV, respectively, when \( \alpha \) is sufficiently large.
\( \eta_{\alpha|d} \) and \( \sigma_{\alpha|d}^2 \) denote the mean and variance of the number of available MASs seen by \( d \)th device, respectively, and are
\[ \eta_{\alpha|d} = L - \eta_B - \sum_{i=1}^{d-1} \eta_Z(\alpha(i,N_z)) \]
\[ \sigma_{\alpha|d}^2 = \sigma_B^2 + \sum_{i=1}^{d-1} \sigma_Z^2(\alpha(i,N_z)). \]  
(25)

As can be seen from Eqs. (23) and (24), \( P_\alpha(\alpha|d,N_z) \) depends on \( P_\alpha(\alpha|d,N_z) \) for \( \delta = 1, \ldots, d - 1 \). So, we start from \( d = 1 \) and go forth toward \( d = N \) and compute \( P_\alpha(\alpha|d,N_z) \) and \( P_Z(Z(\alpha)|d,N_z) \) recursively, for different values of \( \alpha \) from 0 to \( L - 2 \) (The minimum value of BPL is 2, so the maximum value of \( \alpha \) is \( L - 2 \)).

As the final note in this section, we emphasize that we used the following approximation in our model:
\[ \mathcal{E}\{T(q_n)\} \approx \mathcal{E}\{T(q_n)\}, \]  
(26)
where \( \mathcal{E}\{\cdot\} \) takes the average of its argument, \( T \) denotes throughput and \( q_n \) is defined in appendix, available in the online supplemental material. More precisely, instead of computing the throughput, \( T \), in terms of coordinations \( (X,Y) \) and then taking its average over \( X \) and \( Y \), we first
took the average of \( q_x \) over \( X \) and \( Y \) and then computed \( T \) based on the average value of \( q_x \). However, the exact value for \( \mathbb{E}(T(q_x)) \) is

\[
\mathbb{E}(T(q_x)) = T(\mathbb{E}(q_x)) + N\sigma^2 + \cdots
\]

where \( \sigma^2 \) denotes the variance of \( q_x \) and \( \mu_n = \mathbb{E}(\{q_n - \mathbb{E}(q_n)\}^n) \) is the central moment of \( q_x \). However, we used only the first term of Eq. (27) in Eq. (26) which is an approximation. However, computing the exact value leads to a set of complicated equations which should be solved using numerical methods.

To compute fairness and starvation in the network, in the following sections, we propose two objective formulas.

### 3.4 Fairness

A MAC protocol is fair if all devices in the network can use the network evenly. In almost all current MAC protocols, like IEEE 802.3, IEEE 802.11, IEEE 802.15, etc., the media access is designed to be fair. There exist a good record of works in the literature trying to provide a quantitative measure for fairness [18], [19], [20], [21], [22]. However, these measures were complicated and/or could not satisfy the following requirements:

- In a network containing \( N \) devices, if only one device can use the whole media and others achieve zero throughput, the fairness is zero.
- If all devices have equal throughput, the fairness is 1.
- If \( N' \) devices have equal throughput of \( \tau \) and the other ones have zero throughput, the fairness is \( \frac{N' - 1}{N} \).

We define this to satisfy a linear relation between the concept of fairness and its numerical value.

To have a general formula, we should eliminate \( \tau \) and \( N' \) from the above equations. Therefore, we use empirical mean and variance of the throughput. When \( N' \) devices have equal throughput of \( \tau \), and other ones have zero throughput, the empirical variance would be

\[
s = \frac{1}{N - 1} \left( \sum_{i=1}^{N'} (\tau - \bar{\tau})^2 + \sum_{i=N'+1}^{N} (0 - \bar{\tau})^2 \right) = \frac{N'(N - N')}{N(N - 1)} \tau^2
\]

in which \( \bar{\tau} = \frac{N'}{N} \tau \) denotes empirical mean. As stated before, we define the fairness as:

\[
\mathcal{F} \triangleq \frac{N' - 1}{N - 1}.
\]

Defining \( u \triangleq n / \bar{\tau} \) and substituting \( s \) from (28), we have:

\[
u = \frac{N(N - N')}{N'(N - 1)}.
\]

Solving Eq. (30) for \( N' \) and substituting it in Eq. (29) and with some manipulations, the following formula is obtained for the fairness:

Since \( N \) and \( u \) are always positive, the denominator of Eq. (31) is always positive. Therefore, the fairness will be negative if \( u > N \). This happens when \( s > N \bar{\eta} \). But because \( N \geq 2 \), \( s \) maximizes when one of devices has throughput of \( \tau \) and the other has zero throughput. In this case, \( s = \frac{2\tau}{\bar{\tau}} \) and \( \bar{\eta} = \frac{1}{2} \). Therefore, the fairness will never be negative. Similarly, \( \mathcal{F} > 1 \) when \( \frac{N-1}{2(N-1)} > 1 \) and with some manipulations, we obtain \( uN < 0 \) which is impossible. Hence, always \( 0 \leq \mathcal{F} \leq 1 \).

### 3.5 Starvation

Another important parameter of a MAC protocol is starvation. We define it as the percentage of devices that get zero throughput. Therefore,

\[
S = P_T(T = 0).
\]

This probability can be computed easily from the PMF of throughput given in Eq. (20).

### 4 Optimization of Throughput

A common question arising after computation of throughput in the networks is to find optimum parameters to maximize throughput. Here, we compute the optimum payload size as an example to emphasis on the usefulness of the proposed model. Similar optimization problems now can be solved for other controllable parameters. We solve this problem for a device in operation which it knows its BPL and the number of reserved slots. We start from Eq. (1). In this equation, \( f_s(Z) \) is an increasing function of \( f(Z) \). Therefore, in the optimization process, we can replace it with \( f(Z) \). Considering Eqs. (5) and (11)-(19), we arrive at a simplified form for \( f(Z) \) replacing all ceiling and flooring operators with parentheses and ignoring second part of the right side of Eq. (5) for the sake of simplicity:

\[
f(Z) \sim \frac{\theta_3}{\theta_1 + \theta_2 n_{PL}},
\]

where \( \theta_1 = T_{ACK} + 2 T_{SIFS} + 8 T_n h + n_{FSS} + n_{ATS} \), \( \theta_2 = (8 \times 240/224) T_b N_{agg} \) and \( \theta_3 = (Z \times T_{MAS} - T_{SIFS}) N_{agg} \) are values independent of \( n_{PL} \). Referring to Eq. (1), the new throughput formula to be optimized is therefore:

\[
\tau \simeq \theta_1 \theta_2 n_{PL} - \theta_1 n_{PL} - n_{agg} \rho \theta_2 n_{PL} (1 - \rho)^{8(n_{agg} \theta_1 + n_{agg} \theta_2)}
\]

in which \( \theta_1 = 8/T_{SIFS} \) is another value independent of \( n_{PL} \) and we replaced \( f_s(Z) \) with \( f(Z) \). Now, for the sake of simplicity, we take the logarithm of Eq. (34) and differentiate it with respect to \( n_{PL} \) and equate the final result to zero as follows:

\[
\frac{d \log \tau}{d n_{PL}} = \frac{1}{n_{PL}} - \frac{\theta_2}{\theta_1 + \theta_2 n_{PL}} + 8 \log (1 - \rho) = 0.
\]
We proceed by this example: R is going to reserve some MASs to send some data frames to T. Consider two cases shown in Fig. 4: (a) when R’s beacon is before the T’s beacon and (b) vice versa. In both cases, R requests for its intended MASs in current SF (SF0) and reservation and transmission occurs in the next SF (SF1). So, for current SF, the request should be sent in the previous SF, i.e., at least one SF lag is always exists between request and transmission. In both cases, R sets ReqOwner and ReqTarget of the intended MASs in its DRP IE in SF0 to R and T, respectively. In case (a), after receiving this request, T in response sets ReqOwner and ReqTarget of the MASs requested by R in its DRP IE in SF0 to R and T, respectively, if it is going to accept the request. In the SF1, R sets Owner and Target of the intended MASs in its DRP IE to R and T, respectively and sets Reservation Status to reserved, after which T sets Owner and Target of the same MASs in its DRP IE to R and T, respectively and sets Reservation Status to reserved. After which, transmission takes place in SF1.

In case (b), after receiving R’s request in SF1, T sets Owner and Target of the MASs requested by R in its DRP IE to R and T, respectively and sets Reservation Status to reserved. R then sets Owner and Target of the same MASs in its DRP IE in the SF1 to R and T, respectively and sets Reservation Status to reserved. Finally, transmission takes place.

In FDRP, another parameter, NSF, can be used to announce that this reservation is valid for NSF continuous SFs so that if the other devices can wait until the end of transmission, do not contend for the reserved MASs.

If a third device is going to start a new transmission, it checks whether free MASs are available. If it could not find free MASs, it tries by checking its generated priority against that of current sender. If its generated priority is greater than that of the current sender, it also announces its reservation on its required MASs with the same method described here. The owner of any reservation in requesting for the next SF, when faces such conflicts, checks its own generated priority against the conflicting device, and if its priority is smaller, it ignores its own request and tries in another SF. The target of any device is also checks for such conflicts, and denies any request with less priority with respect to other requesting devices.

A device shall try to reserve MASs in the following preference order:

1) MASs it reserved as owner in previous SF for transmission in current SF.
2) Free (available) MASs.
3) MASs a neighbor device reserved as owner in current SF or is going to reserve for the next SF.
4) MASs for which the device itself is target in current SF or is going to be target in next SF.
5) MASs for which a neighbor device is target in current SF or is going to be target in next SF.

A device at first checks for type 1 MASs, and if it could not find any, or if it needs more MASs, it looks for type 2 and so on. This ensures that if there exists available MASs, the devices do not contend for reserved MASs of other devices.

Table 2 and Fig. 5 show an example of FDRP operation. Fig. 5 depicts the DRP IE of four devices A, B, C and D which are working in their vicinity and all of them sense each other. A sends to B and D to C. In SF0, A is sending to B and announces the MASs shown in the first row of Table 2 as reserved and also tries to reserve the same MASs for SF1. B also announces that it is currently receiving from A and it is going to receive from A in SF1 (second row of Table 2). D enters in contention with A by producing a priority of 574 (fourth row of Table 2). Since, A generated 473 as its priority, it loses in SF0. Therefore, in SF1 A ignores its intention on reservation by setting the mentioned MASs Owner and Target to NaN. However, A tries for reserving the same MASs for SF2 by setting ReqOwner and ReqTarget values appropriately. As a result, B again ignores its request in SF2 and tries for SF3. C again responds to D’s request in SF2 and both C and D announce the MASs as reserved in SF2. However, since D generated smaller priority in SF2, it does not set ReqOwner and ReqTarget values in this SF, and so A can reserve the MASs in SF3.

On the other hand, C in response to D sets the requested MASs Owner and Target to D and C, respectively, in SF1 and its status to reserved, as the transmission is going to be done in SF1. After receiving C’s response, D sets its parameters appropriately and starts its transmission in SF1 in the reserved MASs. Furthermore, D tries to take part in contention for the same MASs to reserve them for SF2, as it generates a greater priority in SF1 with respect to A. So it sets ReqOwner and ReqTarget values appropriately. Thereafter, A again ignores its request in SF2 and tries for SF3. C again responds to D’s request in SF2 and both C and D announce the MASs as reserved in SF2. However, since D generated smaller priority in SF2, it does not set ReqOwner and ReqTarget values in this SF, and so A can reserve the MASs in SF3.

Note that, while in DRP no method is provided to apply priority to some devices over other ones, in FDRP one can do this by allowing a device to generate larger values of random priority.

As another example, consider this scenario: five devices with omni-directional antennas are working in vicinity, all in antenna range of each other. They are programmed to send their traffic as follows: 1 → 4, 2 → 5, 3 → 1, 4 → 1, and 5 → 2. Fig. 6a shows the reserved MASs by each device in the current definition of standard. The vertical axis shows the number of reserved MASs as owner by each device as a function of time (SF) (a) DRP; (b) DRP + relinquish request; (c) FDRP.

As it can be seen, Dev 1 and 4 could obtain almost half of available MASs, but a small fraction of MASs is reached to Dev 3. The share for Dev 2 and 5 was zero. This reflects the unfair behavior of the standard. Fig. 6b shows the same diagram for when devices can use relinquish request. Note that while more devices could reserve MASs, still one device obtained nothing. Also, the reservations are not equal. In Fig. 6c the result of the same scenario with FDRP is shown. As it can be seen, all five devices could randomly reserve their needed MASs, one by one.
Note that in Fig. 6c, some few SF are not used by any device. While these are more occurring in initial SFs and gradually they disappear, they are inevitable at all. Also, note that between two SFs where Dev 4 could acquire the channel, a delay of about 880 ms exists. These are the prices we pay for achieving fairness. However, in Figs. 6a and 6b, some devices could not acquire the channel, and their delay would be infinity!

The FDRP method has many more details, which because of lack of space we bypass them. It is developed in our simulator and checked against any arisen situations and now works correctly.

Following, we discuss how to update our proposed model to be able to compute the throughput of FDRP. In this case, in the steady state, the number of reserved MASs is no longer a function of $d$, as it does not matter who is starting to reserve first. More precisely, when the number of frames pushed to queues of devices is less than the channel capacity, the protocol acts like regular DRP and the device reserves $f^{-1}(f_{in} + f(Z) - f_{out}(Z))$ frames. Otherwise, $L - B$ available MASs are equally distributed among $N_2 + 1$ devices in each device's neighborhood. Therefore,

$$Z = \begin{cases} \frac{L - B}{N_2 + 1}, & f_{in} + f(Z) - f_{out}(Z) \geq f_{out}(Z) \\ f^{-1}(f_{in} + f(Z) - f_{out}(Z)), & \text{o.w.} \end{cases}$$

(38)

So, we can compute the PMF of $Z$ by:

$$P_Z(Z|N_2) = P_B(L - (N_2 + 1)Z|N_2).$$

(39)

Now, we can compute $P_Z(Z)$ using the total probability theorem as:

$$P_Z(Z) = \sum_{N_2=1}^{N-1} P_Z(Z|N_2) P_{N_2}(N_2).$$

(40)

These are the only required changes to the model and the other formulas from (1)-(21) remain unchanged.

### 6 Experimental Results

We implemented a simulator using C++ to evaluate our analytical models. We named it BeeSim (a word in Persian meaning wireless). The parameters set in our simulations are taken from mode A2 of ECMA-387. The data-rate in this mode is given as 1.588 Gbps in the standard. Note that the maximum achievable throughput in this mode is about 1.485 Gbps when only two devices exist in the network and only one of them sends frames with payloads of size 58,324 octets with $a = 0$ and $\rho = 0$.

We used the default values given in Table 3 unless explicitly specified. Also, in all simulations we used $r = 10$ m. Each experiment is repeated 100 times to find reliable results. The duration of simulations is set to 1,000 SFs.

Fig. 7 shows the average throughput, starvation, and fairness of the three protocols vs. different parameters. The solid lines correspond to the proposed analytical models and the dashed lines to simulations. The values of average throughput are in Mbps and are read from the left vertical axis. The starvation and fairness are in range $(0, 1)$ and are read from the right vertical axis. We omitted some of the curves to make the figures less crowded.

In Fig. 7a, the average throughput of DRP ($\bigcirc$), DRP with relinquish request ($\bigtriangleup$), and FDRP ($\times$); and average fairness of DRP ($\bigodot$), DRP with relinquish request ($\bigstar$), and FDRP ($\bullet$) are plotted versus payload size, $n_{pl} = 64, \ldots, 25,600$ bytes. All other values are taken from Table 3. For all three protocols, increasing payload size leads to an increase in the average throughput up to a maximum and then it decreases from that point as expected, because small values of payload size decrease the ratio between payload and header and large values of payload increase frame error rate. Note also that while in all protocols, the average throughputs are almost the same, the fairness of FDRP is near 1, while DRP is less fair when frame size is large. The relinquish request could not completely solve the fairness problem of DRP, as expected.

Fig. 7b depicts the average throughput and fairness of DRP and FDRP along with their starvation (DRP $\bigcirc$ and FDRP $\bullet$) vs. input load, $\lambda \times n_{pl} = 82.4, \ldots, 1,648$ Mbps. As it can be seen, at first, the average throughput increases linearly up to a threshold and then remains constant. However, in DRP the starvation remains almost constant with the value of about zero before the saturation of the throughput and then increases up to 0.55 while in FDRP the starvation remains almost constant at 0. Similarly, the fairness of DRP decreases in saturation to around 0.4 while the fairness of FDRP remains almost perfect.

In all other figures, we only depicted the average throughput and fairness of DRP and FDRP. Fig. 7c shows the mentioned parameters versus different values of $N = 5, \ldots, 100$. It is evident that throughput and fairness remain almost unchanged with the number of devices. This reveals the fact that when directional antennas are in use, increasing the number of devices in typical ranges has no significant effect on the average throughput. The fairness of all three protocols in this case is about 1 which indicates that all devices can have almost equal share of channel. The results of FDRP are similar to those of DRP which were due to the fact that in the cases where the device has a few neighbors and the input load is light, DRP affords all traffic requirements almost equally.

The horizontal axis in Fig. 7d denotes spacial dimension size, $D = 5, \ldots, 100$ meters. As expected, the average throughput of all protocols increase with dimension size up to a threshold. In small dimension sizes where DRP acts poor, the fairness in FDRP is higher. This improvement achieved in the other plots wherever DRP made small fairness values. Another point that can be noted in this plot and
IEEE Proof all other plots of Fig. 7: while the data-rate is expected to be around 1.5 Gbps, however in these plots the data-rate reached at most up to about 400 Mbps. This has two sources:

1) In fact, the bandwidth would be shared between a device and its SONs. The default settings in Table 3 produces about 2.5 SONs on average for each device. Therefore, each device acquires at most $\frac{1588}{3.5} = 354$ Mbps.

2) The operation of devices in their neighborhood makes some MASs to be unavailable to other devices. Increased number of such MASs, causes the average throughput to be even less than that of computed in clause 1.

However, when a device only sends to another device in an environment with no other device in vicinity, a throughput near 1.5 Gbps is achievable. This is justified by Fig. 3, where the maximum throughput is reached near 1.4 Gbps when BER is about $5 \times 10^{-7}$.

Fig. 7e depicts three mentioned parameters vs. $r = 0, \ldots, 5.1 \times 10^{-4}$. The average throughput rapidly decreases down to zero as expected. DRP’s fairness in this case abruptly reaches to about 0.4 for large values of BER. In such circumstances, when a device cannot send its frames, it reserves more MASs and causes the other devices to acquire less MASs. However, the fairness of FDRP are improved significantly with respect to DRP.

Fig. 7f depicts the mentioned parameters with respect to ADR, $a = 0, \ldots, 100$ times per second. The ADR has significant effect on average throughput and for $a > 40$, the average throughput would be about 0. FDRP uses a pipelining in reserving the MASs, i.e., the devices should try to reserve the MASs one SF earlier. Increasing ADR significantly affects this pipeline and the throughput is more severe than
The accuracy of the proposed model in computing all parameters is evident from Fig. 7 and Table 4. This supports the whole method of modeling and also shows that the approximations and simplifying assumptions used in our method have no significant effect on final results.

To compute the average error of our model, we used the following formula:

$$\text{Err} = \frac{T_{\text{avg,Ana}} - T_{\text{avg,Sim}}}{1.588 \times 2^{10}} \times 100\%,$$

in which, the denominator is the defined value of data-rate for mode A2 of ECMA-387 standard to have a constant reference. The average absolute error for all plots of Fig. 7 is written in Table 4 and is less than 2 percent which means that the proposed model is precise enough. Also, our model slightly underestimates the true throughput on average.

### 7 Conclusion

Throughput of the DRP protocol for UWB and 60 GHz mmWave wireless networks based on ECMA-368 and ECMA-387 standards are computed analytically and evaluated using a simulation environment written in C++. We also computed the optimum payload size to achieve maximum throughput. Then we proposed formulas to quantify fairness and starvation. Results reflect this fact that in some conditions, the protocol is unfair. To resolve this, an enhanced version of DRP, called FDRP is also developed and the required changes to the model of DRP to cover FDRP are presented. Results show that while FDRP has no significant effect on average throughput, it’s fairness is almost perfect and its starvation is near zero. Average errors of the analytical models are reported to be less than 2%, which reflects their accuracy.

### References


