

# CARA-OHT: Collision-Aware Rate Adaptation for Optimal High-Throughput in IEEE 802.11s Wireless Mesh Networks

<https://doi.org/10.3991/ijim.v14i13.13245>

Teuku Yuliar Arif <sup>(✉)</sup>, Munandar, Ramzi Adriman, Rizal Munadi  
Universitas Syiah Kuala, Banda Aceh, Indonesia  
yuliar@unsyiah.ac.id

**Abstract**—The wireless mesh network (WMN) is a future network technology that develops single-hop wireless local area networks (WLANs) into multi-hop wireless mesh networks, based on the IEEE 802.11s standard. However, this development still presents many challenges, such as determining the best route between sources and destinations, especially taking into account the use of the medium access control (MAC) and physical (PHY) layers of IEEE 802.11n/ac. Some papers have proposed rate adaptation algorithms particularly for single-hop networks; however, these only focused on mechanisms to find data rates suitable for channel conditions. In IEEE 802.11s WMNs, the design of the rate adaptation algorithm is more challenging. Besides considering the problem of channel quality and optimal throughput, it is necessary to also consider the problem of collision and the best route. It is important to take collision into account because the collision probability in multi-hop mesh networks is higher than that in single-hop networks and can lead to a lower throughput. Rate adaptation algorithms for IEEE 802.11s WMNs have been proposed in other papers, but they also do not consider the use of the MAC and PHY layers of IEEE 802.11n/ac. In this paper, we propose the collision-aware rate adaptation for optimal high-throughput (CARA-OHT) algorithm for WMN IEEE 802.11s. An evaluation through the extensive use of a network NS-3 simulator shows that the proposed algorithm exhibits a higher throughput 5~10% than previously developed algorithms.

**Keywords**—CARA-OHT, Rate Adaptation, Collision-aware, Optimal High Throughput, 802.11s WMN.

## 1 Introduction

The wireless mesh network (WMN) is a future network technology that develops a single-hop wireless local area network (WLAN) into a multi-hop wireless mesh network. Development of a single-hop network of IEEE 802.11 WLANs into multi-hop wireless mesh networks has been established based on the IEEE 802.11s standard document [1]. However, the development of WMNs to date still presents many challenges, such as determining the best route between sources and destinations, in particular, by considering the development of medium access control (MAC) and physical (PHY)

layers in IEEE 802.11n [2] and IEEE 802.11ac [3]. Determining the best route for IEEE 802.11s WMNs must be able to take advantage of developments in the MAC and PHY layers of IEEE 802.11n/ac WLAN so that optimal mesh end-to-end network throughput can be generated.

In IEEE 802.11s WMNs, the best route is determined based on the hybrid wireless mesh protocol (HWMP) routing algorithm. However, the best route in HWMP only uses basic criteria in the form of Airtime metrics and number of hops. Airtime metrics are measured based on channel access overhead, PHY overhead, data rate, and packet error rate. While the number of hops is the number of mesh intermediate nodes that are passed between the sources to the destination. Determining the best route by HWMP does not consider the end-to-end throughput between source and destination.

Meanwhile, IEEE 802.11 WLANs have experienced rapid development in both the MAC and the PHY layers. The IEEE 802.11n MAC provided an aggregate MAC protocol data unit (AMPDU) scheme to improve the efficiency of PHY layer headers. The IEEE 802.11n MAC also proposed the enhanced distributed channel access (EDCA) method, which is used to provide different quality of service (QoS) levels for multimedia traffic. The IEEE 802.11n PHY provided 20/40 MHz channel bandwidth, 800/400 ns guard interval (GI), and multiple input multiple output (MIMO) transmission systems with up to four spatial streams [2], [27]. Based on these developments, the PHY of IEEE 802.11n has 128 data rate options ranging from 6.5 Mbps to 600 Mbps. However, each choice of data rate will produce throughput based on different transmission distances. Therefore, the IEEE 802.11 MAC is also available with a rate adaptation algorithm to determine the optimal data rate for data transmission according to the transmission distance.

In some papers, a rate adaptation algorithm has been proposed especially for single-hop networks of IEEE 802.11n/ac WLANs. Most of the rate adaptation algorithms [4-11], focus on the mechanism needed to find the data rate that matches the channel conditions. In WMNs, the design of the rate adaptation algorithm is more challenging. In addition to considering the problem of optimal distance and throughput, it must also consider the problem of packet collision and the best route. Collision issues are important to take into account because the probability of collisions in multi-hop mesh networks is higher than that in single-hop networks. If the probability of collision is high, it can reduce throughput. On the other hand, the rate adaptation algorithm in WMNs also needs to work cross-layer together with the HWMP routing algorithm to determine the best route, while producing optimal end-to-end throughput.

In some literature, rate adaptation algorithms have also been proposed that work cross-layer with the HWMP routing algorithm. Based on literature studies, researchers in [12] were the first to propose the cross-layer method. This group [12] used statistical data provided by routing metrics to calculate the optimal data rate. The optimal choice of data rate was then used to calculate more accurate link cost estimates. Paper [13] improved routing by considering QoS in HWMP to guarantee end-to-end delay and throughput in mesh networks. Paper [14] considered the effect of the position of relative links in end-to-end routes when determining data rates to avoid dense areas. Paper [15] considered the effect of Airtime ping-pong on the rate adaptation algorithm and HWMP routing algorithm. Paper [16] used the queue analysis method to study the trade-off

between data rates, end-to-end paths, and network contention. Paper [17] proposed joint link scheduling and routing through a cross-layer scheme for IEEE 802.11s WMNs. However, the cross-layer method used in some papers [12-17] did not consider the development of the IEEE 802.11n/ac MAC and PHY in determining the choice of data rate, nor the best route to produce optimal throughput in IEEE 802.11s WMNs.

Paper [18] proposed an algorithm called Collision Aware Rate Adaptation for High Throughput (CARA-HT). In this case, the rate adaptation algorithm considers the problem of collision in the network when determining the data rate that will be used to transmit data from source to destination. The CARA-HT algorithm considers the AMPDU scheme, the choice of channel bandwidth, guard interval, and the number of spatial streams in IEEE 802.11n to determine the appropriate data rate choice. Simulation results show that the CARA-HT algorithm was successful at adapting rates based on channel conditions and collision conditions in single-hop networks. However, the resulting throughput is not optimal, especially at the beginning of transmission and when searching for the optimal data rate when changes in channel conditions occur. The CARA-HT rate adapting capability based on collision conditions in networks is very suitable for use in multi-hop mesh networks to reduce the probability of collision in IEEE 802.11s WMNs. However, CARA-HT was designed only for single-hop WLAN networks.

Based on the background and the problems that we have described before, in this paper we propose optimization of the CARA-HT algorithm to produce optimal throughput based on channel conditions, and when packets collide, optimal throughput for WMN IEEE 802.11s. In this paper we also compared the performance of the CARA and Constant Rate algorithms in multi-hop mesh networks (IEEE 802.11s) using the NS-3 network simulator [19, 20]. The evaluation results show that the proposed algorithm has better performance than the comparison algorithm does.

The rest of this paper is organized as follows. Section II describes previous work related to the proposed rate adaptation algorithm for the IEEE 802.11s standard. Section III describes our proposed algorithm. Section IV describes the simulation method used to evaluate the performance of our proposed algorithm. Section V describes the evaluation results and discussion. Section VI presents the conclusions.

## 2 Related Works

In some literature, rate adaptation algorithms have been proposed that work together using the cross-layer method with the HWMP routing algorithm. Metric-Aware Rate Adaptation (MARA) algorithm has been proposed in [12] that used a cooperative cross-layer method to automate selection of data rates based on routing metrics. In the MARA algorithm, the data rate is determined based on the statistics provided by the routing metrics. The selected data rate is then used for more accurate link cost-estimation calculations. MARA was developed in two variants, namely MARA-P and MARA-RP. In MARA-P, the size of each packet is taken into account when determining the data rate for packet transmission. Whereas in MARA-RP, in addition to the size of the packet

calculated for determination of the data rate, the packet size is also used to determine packet transmission routes.

A reactive phase in HWMP by including transmission delay, traffic load, and bandwidth in calculating routing metrics to match the needs of end-to-end QoS multimedia services has been developed in [13]. The effect of relative link position on end-to-end routes when determining data rates has been considered in [14]. Metric routing calculations are performed using a new metric called expected transmission cost (ETM) in multi-rate wireless networks. The ETM metric with a greedy routing algorithm can be used to determine the best end-to-end route in the multi-hop mesh network. ETM metrics can also be used to determine dynamically the best transmission data rate for each path on the route.

In [15] showed that the basic metric used for calculating the best route in multi-hop mesh networks caused throughput instability (called the ping-pong effect). From the analysis carried out, the ping-pong effect also affects the rate adaptation algorithm. To overcome the ping-pong effect, in [15] proposed a ping-pong-aware algorithm that can be used to detect networks that experience ping-pong effects. The ping-pong-aware algorithm is then used to calculate routing metrics to increase IEEE 802.11s WMN throughput.

The hop-interference trade-off-based rate adaptation for mesh (HITRAM) algorithm has been proposed in [16] using the queue analysis method to study the trade-off between data rates, length of end-to-end routes, and contention in the network. Researchers in [17] proposed a load balancing routing method to maximize the minimum available resource under the given traffic pattern and scheduling method to achieve maximal link utilization on the given route. The two methods were iterated until an optimized solution could be obtained. The proposed algorithm can be formulated using a mathematical LP problem and they showed that it was very effective for load balancing compared to the simple adoption of IEEE 802.11s (a standard TDMA protocol in WMNs). Other researchers in [24] proposed a new method of channel selection in multi-radio multi-channel WMNs to improve the user performance in downlink direction of Internet traffic. However, the cross-layer method used in the literature [12-17], [25] and [26] does not consider the development of IEEE 802.11n/ac MAC and PHY in determining the choice of data rate, nor the best route to produce optimal throughput in IEEE 802.11s WMNs.

Researchers in [18] improved the collision-aware rate adaptation (CARA) algorithm to use the advantages of MAC and PHY 802.11n capabilities and called it CARA for high throughput (CARA-HT). They added a function for checking the success or failure of MPDU transmission using the AMPDU, DataTxVector, and RtsTxVector checking functions based on bandwidth of channel, guard interval and number of streams, and checking function of data transmission requirements using RTS/CTS mode. They implemented the CARA-HT algorithm in NS-3.26. They also evaluated the performance of the CARA-HT algorithm. The simulation results show that the throughput of CARA-HT is better than that of Minstrel-HT.

### 3 Proposed CARA-OHT Algorithm

Based on literature studies, many papers have proposed rate adaptation algorithms. However, most of the proposed algorithms are designed and evaluated for older IEEE 802.11b/a/g WLAN standards. Several recent algorithms have been proposed for rate adaptation in the IEEE 802.11n/ac High Throughput WLANs. However, these algorithms also were not evaluated for use with IEEE 802.11s WMNs. In this section, we describe our proposed rate adaptation algorithm, which works based on the group rate, as shown in Table 1. An abbreviation of the algorithm is shown in Table 2.

**Table 1.** Data rate of IEEE 802.11n WLANs

MCS	Modulation	R	Nss	20 MHz						40 MHz							
				GI			SGI			GI			SGI				
				tr	gl	rl	dr	tr	gl	rl	dr	tr	gl	rl	dr	tr	gl
0	BPSK	1/2	1	0	0	6.5	32	0	7.2	64	0	13.5	96	0	15	0	15
1	QPSK	1/2	1	1	1	13	33	1	14.4	65	1	27	97	1	30	1	30
2	QPSK	3/4	1	2	2	19.5	34	2	21.7	66	2	40.5	98	2	45	2	45
3	16-QAM	1/2	1	3	3	26	35	3	28.9	67	3	54	99	3	60	3	60
4	16-QAM	3/4	1	4	4	39	36	4	43.3	68	4	81	100	4	90	4	90
5	64-QAM	2/3	1	5	5	52	37	5	57.8	69	5	108	101	5	120	5	120
6	64-QAM	3/4	1	6	6	58.5	38	6	65	70	6	121.5	102	6	135	6	135
7	64-QAM	5/6	1	7	7	65	39	7	72.2	71	7	135	103	7	150	7	150
8	BPSK	1/2	2	8	0	13	40	0	14.4	72	0	27	104	0	30	0	30
9	QPSK	1/2	2	9	1	26	41	1	28.9	73	1	54	105	1	60	1	60
10	QPSK	3/4	2	10	2	39	42	2	43.3	74	2	81	106	2	90	2	90
11	16-QAM	1/2	2	11	3	52	43	3	57.8	75	3	108	107	3	120	3	120
12	16-QAM	3/4	2	12	4	78	44	4	87.7	76	4	162	108	4	180	4	180
13	64-QAM	2/3	2	13	5	104	45	5	115.6	77	5	216	109	5	240	5	240
14	64-QAM	3/4	2	14	6	117	46	6	130	78	6	243	110	6	270	6	270
15	64-QAM	5/6	2	15	7	130	47	7	144.4	79	7	270	111	7	300	7	300
16	BPSK	1/2	3	16	0	19.5	48	0	21.7	80	0	40.5	112	0	45	0	45
17	QPSK	1/2	3	17	1	39	49	1	43.3	81	1	81	113	1	90	1	90
18	QPSK	3/4	3	18	2	58.5	50	2	65	82	2	121.5	114	2	135	2	135
19	16-QAM	1/2	3	19	3	78	51	3	86.7	83	3	162	115	3	180	3	180
20	16-QAM	3/4	3	20	4	117	52	4	130	84	4	243	116	4	270	4	270
21	64-QAM	2/3	3	21	5	156	53	5	173.3	85	5	324	117	5	360	5	360
22	64-QAM	3/4	3	22	6	175.5	54	6	195	86	6	364.5	118	6	405	6	405
23	64-QAM	5/6	3	23	7	195	55	7	216.7	87	7	405	119	7	450	7	450
24	BPSK	1/2	4	24	0	26	56	0	28.9	88	0	54	120	0	60	0	60
25	QPSK	1/2	4	25	1	52	57	1	57.8	89	1	108	121	1	120	1	120
26	QPSK	3/4	4	26	2	78	58	2	86.7	90	2	162	122	2	180	2	180
27	16-QAM	1/2	4	27	3	104	59	3	115.6	91	3	216	123	3	240	3	240
28	16-QAM	3/4	4	28	4	156	60	4	173.3	92	4	324	124	4	360	4	360
29	64-QAM	2/3	4	29	5	208	61	5	231.1	93	5	432	125	5	480	5	480
30	64-QAM	3/4	4	30	6	234	62	6	260	94	6	486	126	6	540	6	540
31	64-QAM	5/6	4	31	7	260	63	7	288.9	95	7	540	127	7	600	7	600

**Table 2.** The abbreviations used in Table 1

Abbreviation	Explanation
MCS	Modulation and Coding Scheme
R	Coding rate
Nss	Number of spatial streams
GI	Guard interval
SGI	Short guard interval
tr	txrate
gl	Group index
rl	Rate index
dr	Data rate

### 3.1 Optimization of the initial value of the rate index

In the CARA-HT algorithm, the initial index rate always starts with txrate = 0 [18]. This condition causes CARA-HT to take longer to reach its maximum data rate. To overcome this problem, we propose the optimization of the initial value of transmission rate (txrate) as shown in Pseudocode 1. In CARA-OHT, the initial value of txrate is determined by the PHY configuration used. Thus, CARA-OHT begins by getting information about the channel bandwidth (chWidth), guard interval (sgi), and the number of spatial streams (streams) [Pseudocode 1, lines 2-4]. Then the initial value of txrate is determined according to the PHY configuration used [Pseudocode 1, lines 6-21].

**Pseudocode 1.** Initialize the initial value of the data rate of the proposed algorithm

1:	<b>Begin</b>
2:	chWidth = GetChannelWidth ();
3:	sgi = GetShortGuardInterval ();
4:	streams = GetNumberOfSupportedRxAntennas ();
5:	<b>if</b> (totalPacketsCount == 0) <b>then</b>
	<b>if</b> ((chWidth == 20) && (sgi == false) && (streams == 1))
	<b>then</b>
	txrate = 0;
6:	<b>else if</b> ((chWidth == 20) && (sgi == false) && (streams ==
2):	<b>then</b>
	txrate = 8;
7:	<b>else if</b> ((chWidth == 20) && (sgi == false) && (streams ==
3):	<b>then</b>
	txrate = 16;
8:	<b>else if</b> ((chWidth == 20) && (sgi == false) && (streams ==
4):	<b>then</b>
	txrate = 24;
9:	<b>else if</b> ((chWidth == 20) && (sgi == true) && (streams ==
1):	<b>then</b>
	txrate = 32;
10:	<b>else if</b> ((chWidth == 20) && (sgi == true) && (streams ==
2):	<b>then</b>
	txrate = 40;
11:	<b>else if</b> ((chWidth == 20) && (sgi == true) && (streams ==
3):	<b>then</b>
	txrate = 48;
12:	<b>else if</b> ((chWidth == 20) && (sgi == true) && (streams ==
4):	<b>then</b>
	txrate = 56;
13:	<b>else if</b> ((chWidth == 40) && (sgi == false) && (streams ==
1):	<b>then</b>
	txrate = 64;

```

14: 2)) then
        txrate = 72;
        else if ((chWidth == 40) && (sgi == false) && (streams ==
15: 3)) then
        txrate = 80;
        else if ((chWidth == 40) && (sgi == false) && (streams ==
16: 4)) then
        txrate = 88;
        else if ((chWidth == 40) && (sgi == true) && (streams ==
17: 1)) then
        txrate = 96;
        else if ((chWidth == 40) && (sgi == true) && (streams ==
18: 2)) then
        txrate = 104;
        else if ((chWidth == 40) && (sgi == true) && (streams ==
19: 3)) then
        txrate = 112;
        else if ((chWidth == 40) && (sgi == true) && (streams ==
20: 4)) then
        txrate = 120;
21: end

```

### 3.2 Optimization of data rate increment method

The CARA-OHT data rate adaptation design for channel quality improvement is shown in Pseudocode 2. Several new functions are added in CARA-OHT so that data rate adaptation can be optimized based on channel quality improvement. In the CARA-HT algorithm, if MPDU is always successfully transmitted and every threshold of transmission success is achieved, the index rate is incremented in sequence. This arrangement can in certain conditions be detrimental, because when the index rate is increased, the value of the data rate actually decreases, as shown in Table 1.

Therefore, the CARA-OHT design was carried out as follows. If the threshold for the number of transmission successes is reached or if the timer timeout is reached, and the index rate value is smaller than the maximum index rate value in the group rate, the index rate value is incremented [Pseudocode 2, lines 9-34]. If the value of the data rate of the increment index rate is smaller than the value of the previous data rate, then incrementation of the index value is continued until a data rate higher than the previous data rate is found [Pseudocode 2, lines 35-47].

**Pseudocode 2.** Increment method of the data rate of the proposed algorithm

```

1: Begin
2:   rateId = GetRateId (txrate);
3:   groupId = GetGroupId (txrate);
4:   numRates = 8;
5:   numGroups = 16;
6:   timer++; suc-
7:   cess++; failed
8:   = 0;
9:   if ((success == successThreshold || timer ==timerTimeout)
10:    && (txrate < (groupId+1) * numRates)) then
11:     mcsIndex = GetMcsIndek(groupId, rateId);
12:     group = CaraoGroups[groupId];
13:     dataRate = GetDataRate(mcsIndex, groupId(chWidth, sgi,
14: streams));
15:     txrate++;
16:     rateIdNext = GetRateId (txrate);
17:     groupIdNext = GetGroupId (txrate);
18:     mcsIndexNext = GetMcsIndek(groupIdNext, rateIdNext);
19:     dataRateNext = GetDataRate(mcsIndexNext,
20: groupIdNext(chWidth, sgi, streams);
21:
22:     if (txrate == (groupId+1) * numRates) then
23:       nextGroup = groupId;
24:       do
25:         nextGroup++;
26:         nextGroup %= numGroups;
27:         while (!nextGroup.supported);
28:         if (nextGroup != 0) then
29:           rateIdNext = 0;
30:           txrate = GetIndex (nextGroup, rateIdNext);
31:           groupNext = CaraoGroups[nextGroup]; mcsIn-
32: dexNext = GetMcsIndek(nextGroup, rateId); da-
33: taRateNext = GetDataRate(mcsIndexNext,
34: groupNext(chWidth, sgi, streams);
35:
36:         else
37:           txrate--;
38:           rateIdNext = GetRateId (txrate);
39:           mcsIndexNext = GetMcsIndek(nextGroup, rateId);
40:           dataRateNext = GetDataRate(mcsIndexNext,
41: groupNext(chWidth, sgi, streams);
42:
43:         if ( dataRateNext < dataRate) then
44:           do
45:             txrate++;
46:             if (txrate == (nextGroup +1) * numRates) then
47:               nextGroup++;
48:               rateIdInc = GetRateId (txrate);
49:               groupIdInc = GetGroupId (txrate);
50:               mcsIndexInc = GetMcsIndek(groupIdInc, rateIdInc);
51:               groupInc = CaraoGroups[nextGroup];
52:               dataRateInc = GetDataRate(mcsIndexInc,
53: groupInc(chWidth, sgi, streams);
54:
55:             while ( dataRateInc < dataRate );
56:           timer = 0;
57:           success = 0;
58:
59: End

```

### 3.3 Optimization of data rate decrement method

Design optimization of the CARA-OHT data rate adaptation when MPDU fails to be transmitted due to decrease in the channel quality, is shown in Pseudocode 3. In the CARA-HT algorithm, if MPDU always fails to be transmitted and the transmission failure threshold is reached, the index rate is decremented sequentially. With these arrangements, certain conditions also cause losses. This is because lowering the index rate causes the value of the data rate to increase, causing failure of the MPDU transmission to increase.

To overcome this problem, optimization of CARA-OHT was designed as follows. If the transmission failure threshold is reached or the timer timeout is reached, and the index rate value is not equal to zero; then the index value is decremented [Pseudocode 3, lines 6-25]. If the value of the data rate from the decrement index rate is greater than the value of the previous data rate, then the index value is decremented until a data rate is found that is smaller than the previous data rate [Pseudocode 3, lines 26-36].

**Pseudocode 3.** Decrement method of the data rate of the proposed algorithm

1:	<b>Begin</b>
2:	rateId = GetRateId (txrate);
3:	groupId = GetGroupId (txrate);
4:	numRates = 8;
5:	numGroups = 16;
6:	<b>if</b> (txrate != 0) <b>then</b>
7:	mcsIndex = GetMcsIndex(groupId, rateId);
8:	group = AarfcdoGroups[groupId];
9:	dataRate = GetDataRate(mcsIndex,
10:	groupId(chWidth, sgi, streams);
11:	txrate--;
12:	<b>if</b> (txrate == (groupId * numRates)-1)
13:	<b>then</b>
14:	nextGroup = groupId;
15:	<b>do</b>
16:	nextGroup--;
17:	nextGroup %= numGroups;
18:	<b>while</b> (!nextGroup.supported);
19:	rateId = 7;
20:	txrate = GetIndex (nextGroup,
21:	rateId);
22:	groupNext = CaraoGroups[nextGroup];
23:	mcsIndexNext = GetMcsIndex(next-
24:	Group, rateId);
25:	dataRateNext = GetDataRate(mcsIn-
26:	dexNext, groupNext(chWidth, sgi, streams);
27:	<b>else</b>

```

23:         rateId = GetRateId (txrate);
24:         mcsIndexNext = GetMcsIndex(nextGroup,
rateId);
25:         dataRateNext = GetDataRate(mcsIndexNext,
groupNext(chWidth, sgi, streams);
26:         if (dataRateNext > dataRate) then
27:             do
28:                 txrate--;
29:                 if (txrate == (nextGroup +1) * num-
Rates) then
30:                     nextGroup--;
31:                     rateIdDec = GetRateId (txrate);
32:                     groupIdDec = GetGroupId (txrate);
33:                     mcsIndexDec = GetMcsIndex(nextGroup,
rateId);
34:                     groupInc = CaraoGroups[nextGroup];
35:                     dataRateDec = GetDataRate(mcsIn-
dexDec, groupInc(chWidth, sgi, streams);
36:                     while ( dataRateDec < dataRate );
37:             End

```

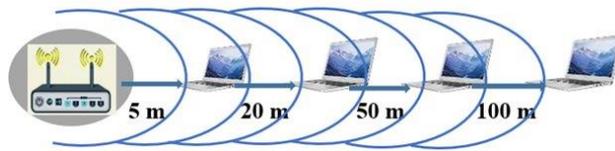
## 4 Simulation Method for CARA-OHT Performance Evaluation

We implemented the CARA-OHT algorithm in a network simulator (NS-3) using C++. We used version NS-3.26, in which the IEEE 802.11n/ac WLAN model is available. This version has been developed and evaluated by other researchers [19, 22, 23]. Next, we carried out evaluation of the CARA-OHT algorithm in three stages. In the first stage, we evaluated the optimization of the index value of the index rate, the index rate increase optimization, and the optimization of the CARA-OHT algorithm index decrease based on the condition of the channel quality change. In the second stage, an evaluation of the CARA-OHT algorithm was carried out based on the collision conditions caused by the hidden terminal. The first and second stage evaluations were carried out on a single hop IEEE 802.11n WLAN. In the third stage, an evaluation of the CARA algorithm was carried out in IEEE 802.11s multi-hop WMNs.

### 4.1 Simulation method for CARA-OHT performance evaluation based on wireless channel quality

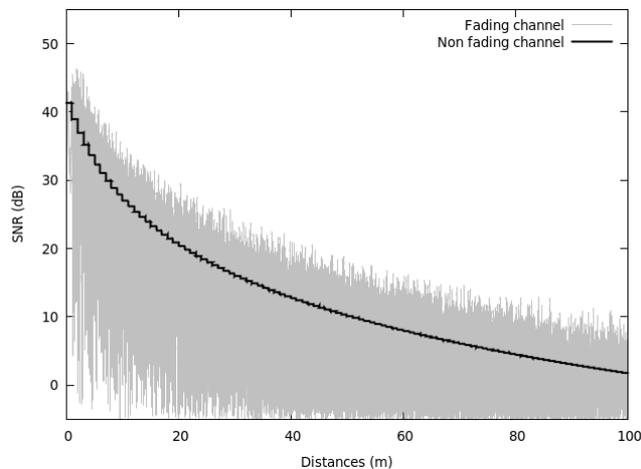
Evaluation of the performance of the CARA-OHT algorithm in relation to changes in channel quality was based on the topology in Figure 1. In the simulation, there were only two nodes, one node as an Access Point (AP) and one node as a station (STA). Both nodes were set to support an IEEE 802.11n WLAN. When the simulation started, the distance between nodes was only 1 m; however, during the simulation the STA

moved away from the AP at a speed of 1 m/s to a distance of 100 m. The movement of STA away from the AP was intended, so that the quality of the channel between the two changed from good (when close together) to worse (when farther apart).



**Fig. 1.** Simulation topology of algorithm performance evaluation based on wireless channel quality

In the simulation, NS-3 was set using the `LogDistancePropagationLoss` channel loss model, the `ConstantSpeed` propagation delay model, and the `NistErrorRateMode` error rate model. Both nodes were set using 1 dBm Tx Power and 1 dB Tx/Rx gain, and the 5 GHz frequency band was used. Other PHY parameters such as channel bandwidth (20/40 MHz), guard intervals (800/400 ns), and the number of spatial streams (2/3/4), were set according to the evaluation scenario. At the MAC layer, both nodes were set to use the CARA-OHT rate adaptation algorithm, to support the A-MPDU scheme, and to use the EDCA scheme best effort category.



**Fig. 2.** IEEE 802.11n channel quality model in 40 MHz channel bandwidth based on distance between AP and STA [21]

During the simulation, when STA was 1 m from the AP, the `OnOffApplication` in STA continuously sent UDP packets to the sink application on the AP. The UDP packets were each 1500 bytes and were sent at the same speed as the maximum PHY data rate based on the configuration used, so that the maximum throughput could be known. As long as the STA moved away from the AP, packets were continuously sent using a data rate that changed according to channel conditions, and where the value of

the data rate used was determined by the CARA-OHT rate adaptation algorithm. The change in the value of the data rate was monitored by STA so that responses to change in the data rate as a result of changes in channel quality could be determined. Packets received in the sink application within a period of one second were summed so that throughput could be known every second. To validate the success of the CARA-OHT algorithm based on channel conditions, simulations were also carried out using the CARA-HT rate adaptation algorithm, so that the performance of both can be compared. Other parameters used in the simulation are shown in Table 6. Changes in the channel quality based on the simulation parameters used, are shown in Figure 2 [21].

#### 4.2 Simulation method for CARA-OHT performance evaluation based on the hidden terminal problem

Evaluation of the CARA-OHT algorithm based on collision conditions due to hidden terminals is based on the topology in Figure 3. In the simulation, in addition to the previous two nodes (as AP and as STA), there was also a third node that caused a hidden terminal. All nodes were set to support IEEE 802.11n WLAN. During the simulation, the STA position was set to be static at a distance of 5 m to the right of the AP, while the STA hidden node was also set at a static distance of 5 m to the left of the AP. To create a collision effect on the AP, the channel model between the AP-STA and the AP-hidden node was set to use the `RangePropagationLoss` channel loss model so that the channel quality in both STAs was equal. In this simulation, the `ConstantSpeed` propagation delay model and the `NistErrorRateModel` error rate model were used. The other PHY and MAC parameters in this simulation were the same as in the first simulation, as shown in Table 3.

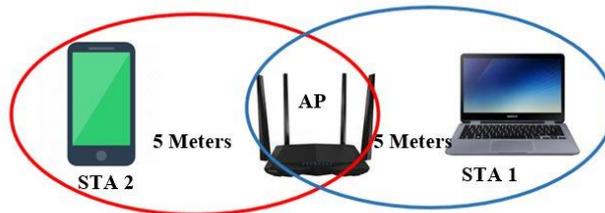


Fig. 3. Simulation topology of CARA-OHT performance evaluation based on wireless channel collision

During the second simulation, `OnOffApplication` in the STA continuously transmitted UDP packets to the sink application in the AP. Similarly, on the side of the STA hidden terminal, `OnOffApplication`, which was in the STA hidden terminal, continuously transmitted UDP packets to the sink application in the AP. Collisions occurred among the UDP packets received by the sink application in the AP. In this simulation, changes in the value of the STA side data rate were monitored so that change in the data rate response by the CARA-OHT rate adaptation algorithm was caused by the collisions. Packets received in the sink application within a period of one second were

summed so that throughput could be known every second during a collision. To validate the success of the CARA-OHT algorithm based on collision, simulations were also carried out using the CARA-HT rate adaptation algorithm so that the performance of both could be evaluated.

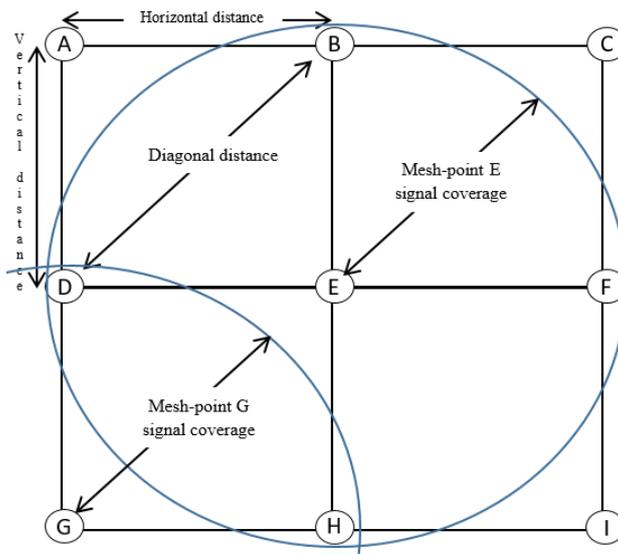
**Table 3.** Simulation Parameters

Parameters	Value
Number of AP	1
Number of STA	1 (1st scenario), 2 (2nd scenario)
Maximum distance STA-AP	100 m (1st scenario), 5m (2nd scen.)
PHY Model	YansWifiPhy
Energy detection threshold	-96.0 dBm
CCA mode threshold	-99.0 dBm
Tx Power	16.02 dBm
Tx/Rx gain	1 dB
Rx noise figure	7 dB
Number of spatial streams	2/3/4
Guard interval	800ns (GI), 400ns (Short-GI)
Channel bandwidth	20/40 MHz
Frequency	5 GHz
Channel loss model	1st scenario: LogDistancePropagationLoss 2nd scenario: RangePropagationLoss
Channel delay model	ConstantSpeed
Error model	NistErrorRateModel
MAC model	HT-MAC
Maximum size of A-MPDU	65535 bytes
RTS/CTS threshold	0 byte (enable RTS/CTS mode) 65535 bytes (disable RTS/CTS mode)
EDCA access category	Best effort
IP network/Transport	IPv4/UDP
Application model	OnOffApplication
Maximum data rate	600 Mbps
Packet size	1500 byte
Appication start time	0.5 s
Simulation time	100.0 s

### 4.3 Simulation method for CARA performance evaluation in WMN IEEE 802.11s

Evaluation of the performance of the CARA algorithm in an IEEE 802.11s WMN was done based on an 802.11s mesh module in NS-3.26 [20]. In the simulation, there were a total of nine nodes that functioned as mesh-points and were placed in 3×3 grid topology, as shown in Figure 4. All mesh-points were set using IEEE 802.11a PHY, which has a range of data rates: 6, 9, 12, 18, 24, 36, 48, and 54 Mbps. During the simulation, the positions of all mesh-points were static. The distance between mesh-points was set to be the same horizontally and vertically so that each could communicate directly only with the mesh-point next to it vertically and horizontally and could not communicate directly with a mesh-point at a diagonal position according to the data rate

used. For example, mesh-point A could only communicate directly with mesh-points B and D but could not communicate directly with mesh-point E and other mesh-points. Thus, if mesh-point A sent packets to mesh-point E or to other mesh-points (not B and D), then the packet had to be routed using the HWMP protocol past mesh-point B or D, to be able to reach its destination. The vertical and horizontal distances between mesh-points based on the data rate used in this simulation are shown in Table 4.



**Fig. 4.** Simulation topology of CARA algorithm performance evaluation in an IEEE 802.11s WMN

In this simulation, the channel model between mesh-points was set using the `Log-DistancePropagationLoss` channel loss model. The propagation delay model was set to use `ConstantSpeed` and the error rate model was set to use `NistError-RateModel`. All mesh-point nodes were set using 1 dBm Tx Power, 1 dB Tx/Rx gain, and the band frequency of 5 GHz. Other PHY parameters were also set: channel bandwidth to 20 MHz, guard intervals to 800 ns, and the number of spatial streams to 1. At the MAC layer, all mesh-point nodes were set using the `ConstantRate` and CARA rate adaptation algorithms, without the A-MPDU scheme, and using the best effort EDCA scheme category.

During the second simulation, the `OnOffApplication` on the C mesh-point continuously sent UDP packets to the sink application in the E mesh-point. Simultaneously, the `OnOffApplication` on the G mesh-point also sent UDP packets continuously to the sink application at mesh-point E. As a result, the probability that the UDP packets received by the sink application at the E mesh-point would cause collisions increased. In this simulation, the change in the value of the data rate at the C mesh-point was monitored so that change in the data rate response by the CARA rate adaptation algorithm was caused by collisions. Packets received in the sink application within a period

of one second were summed so that throughput could be known every second when a collision occurred in the IEEE 802.11s WMN. To evaluate the performance of the CARA algorithm in mesh networks, simulations were also carried out using the `ConstantRate` rate adaptation algorithm so that the performance of both could be compared.

**Table 4.** Diagonal and vertical/horizontal distances between mesh-points based on data rate

Data rate (Mbps)	Diagonal distance (m)	Vertical/ horizontal distance (m)
54	30	21,21
48	35	24,74
36	47	33,23
24	58	41,01
18	78	55,15
12	97	68,58
9	98	69,29
6	120	84,85

## 5 Results and Discussion

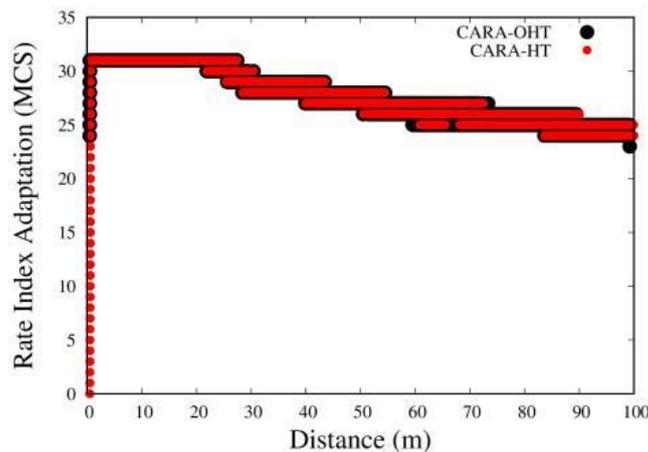
In this section we describe the performance evaluation of the CARA-OHT algorithm. The purpose of the evaluation was to ascertain whether optimization of the initial data rate, optimization of the increase in the data rate for the success of the MPDU transmission, and optimization of the reduction in data rate for MPDU transmission failures in the CARA-OHT algorithm, were functioning properly. A number of functions of the CARA-OHT algorithm were evaluated. These included data transmission functions using and not using AMPDU, data transmission functions using various combinations of the number of spatial streams, guard intervals and channel bandwidth in IEEE 802.11n, and the functions of increasing and decreasing data rates based on channel conditions and collision error to produce optimal throughput.

### 5.1 CARA-OHT performance evaluation based on channel condition

The first simulation scenario run was the CARA-OHT rate adaptation of channel conditions using the topology shown in Figure 1. The PHY parameters used were 20 MHz bandwidth, 800 ns guard interval, and MIMO using 4 spatial streams. Thus, the PHY layer in the AP and STA supported data rates ranging from MCS-0 (6.5 Mbps) to MCS-31 (260 Mbps), as shown in Table 1 (column GI = 800 ns, bandwidth 20 MHz). In this scenario, the AP functions as a recipient of the packets and runs the `ConstantRate` algorithm (`apManager=ns3:ConstantRateWifiManager`). Thus, each packet received without error was confirmed by the AP by transmitting the ACK using a constant data rate. The STA was able to transmit UDP packets (1500 bytes) continuously to the AP using a data rate adapted using the CARA-OHT algorithm (`staManager=ns3::CaraoHtWifiManager`). Packets were transmitted in this simulation scenario without using A-MPDU. To provide comparison to the performance of the

CARA-OHT algorithm, the STA was simulated using both the CARA-HT and the ConstantRate algorithms.

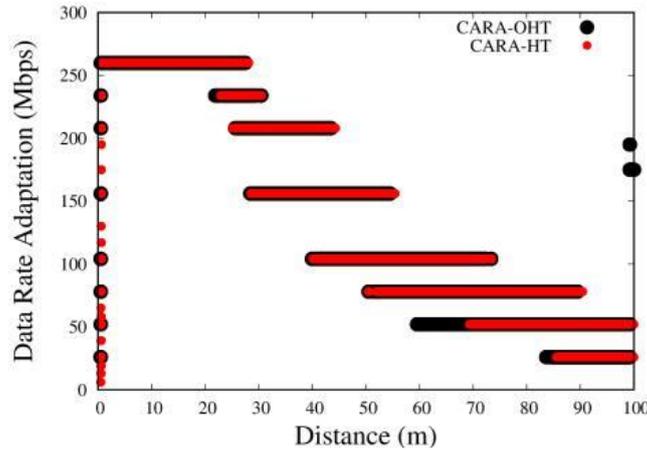
Figure 5 shows the adaptation of the MCS index of the CARA-OHT algorithm to changes in channel conditions. When the distance between the STA and AP was  $< 5$  m, which means that the channel conditions were very good, the initial value of the rate index used by the CARA-OHT algorithm was MCS-24. This initial value is in accordance with the optimization of the initial value index rate designed in Section 3. If every threshold value indicating transmission success was reached, CARA-OHT incremented the index rate until it reached the MCS-31 index. The CARA-HT algorithm was seen using the initial value of the MCS-0 index and then incrementing it to the MCS-31 index. When there was a decrease in channel quality, both algorithms decremented the index rate to MCS-24. Thus, the optimization of the initial value of the MCS index in CARA-OHT was successfully carried out because CARA-OHT reached the highest index rate faster than CARA-HT did, as shown in the detailed simulation results in Table 5.



**Fig. 5.** Adaptation of the index rate of the CARA-OHT and CARA-HT algorithms based on channel conditions (without AMPDU scheme, 20 MHz channel bandwidth, GI = 800 ns and 4 spatial streams)

Figure 6 shows the adaptation of the CARA-OHT algorithm data rate to changes in channel conditions. When the distance between the STA and AP was  $< 5$  m, which means that the channel conditions were still good, the CARA-OHT algorithm was seen using the initial data rate of 26 Mbps. This initial value is in accordance with the design optimization of the initial value index rate designed in Section 3. If every threshold value of the transmission success was achieved, CARA-OHT increased the data rate to reach the highest data rate, which was 260 Mbps. The CARA-HT algorithm was seen using the initial data rate of 6.5 Mbps and then increasing or decreasing the data rate according to the MCS index. Thus, optimization of the initial value of the data rate, and of the method for increasing the data rate on CARA-OHT, was successfully carried out.

That is, there was no decrease in the data rate when continuous channel quality was better than for CARA-HT, as shown in the detailed simulation results in Table 5.



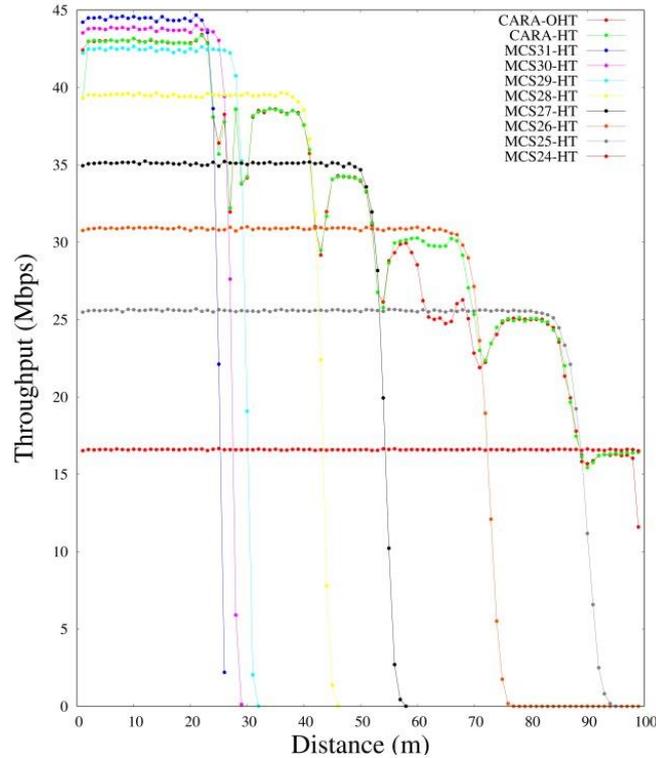
**Fig. 6.** Adaptation of data rate of the CARA-OHT and CARA-HT algorithms based on channel conditions (without AMPDU scheme, 20 MHz channel bandwidth, GI = 800 ns and 4 spatial streams)

Figure 7 shows the CARA-OHT algorithm throughput based on changes in channel conditions. From the figure, it can be seen that the CARA-OHT throughput matches the throughput of ConstantRate according to the optimum distance for each of MCS-24, MCS-25, MCS-26, MCS-27, MCS-28, MCS-29, MCS-30, and MCS-31. When there was no collision in the network, CARA-OHT successfully adapted the data rate according to channel conditions so that the resulting throughput could be optimal according to changes in the channel conditions. The evidence for this optimal throughput is that the graph line shows that the CARA-OHT throughput is similar to that of ConstantRate according to the distance between the AP and STA.

The second scenario of the CARA-OHT rate adaptation simulation of channel conditions was carried out using the same PHY parameters as in the first simulation scenario (bandwidth 20 MHz, 800 ns guard interval, and MIMO 4 spatial stream). The PHY layer in the AP and STA supported data rates ranging from MCS-0 (6.5 Mbps) to MCS-31 (260 Mbps) as shown in Table 1 (columns GI = 800 ns, 20 MHz). In this scenario, the AP also functioned as the recipient of packets and employed the ConstantRate algorithm (ap-Manager=ns3::ConstantRateWifiManager). The STA was enabled to transmit UDP packets of 1500 bytes continuously to the AP using a data rate adapted using the CARA-OHT algorithm (staManager=ns3::CaraoHtWifiManager). In this second scenario, the packets transmitted by the STA to the AP were first aggregated into the A-MPDU frame with a maximum size of 65,535 bytes. In this scenario, the CARA-HT algorithm was used for comparison.

**Table 5.** Comparison of the index/data rate increment and speed reaches the highest index/data rate

CARA-OHT			CARA-HT		
<i>Time (second)</i>	<i>Rate index (MCS)</i>	<i>Data rate (Mbps)</i>	<i>Time (second)</i>	<i>Rate index (MCS)</i>	<i>Data rate (Mbps)</i>
0.503019	24	26	0.503019	0	6
0.50927	25	52	0.520479	1	13
0.514183	26	78	0.531931	2	19
0.517782	27	104	0.53997	3	26
0.521136	28	156	0.546644	4	39
0.523878	29	208	0.551546	5	52
0.526735	30	234	0.556003	6	58
0.529368	31	260	0.560036	7	65
			0.563824	8	13
			0.575321	9	26
			0.582186	10	39
			0.587252	11	52
			0.591614	12	78
			0.595261	13	104
			0.598391	14	117
			0.601653	15	130
			0.604579	16	19
			0.612922	17	39
			0.618559	18	58
			0.622728	19	78
			0.626392	20	117
			0.62932	21	156
			0.632197	22	175
			0.635089	23	195
			0.637951	24	26
			0.644873	25	52
			0.649502	26	78
			0.653119	27	104
			0.656401	28	156
			0.659422	29	208
			0.662189	30	234
			0.664633	31	260

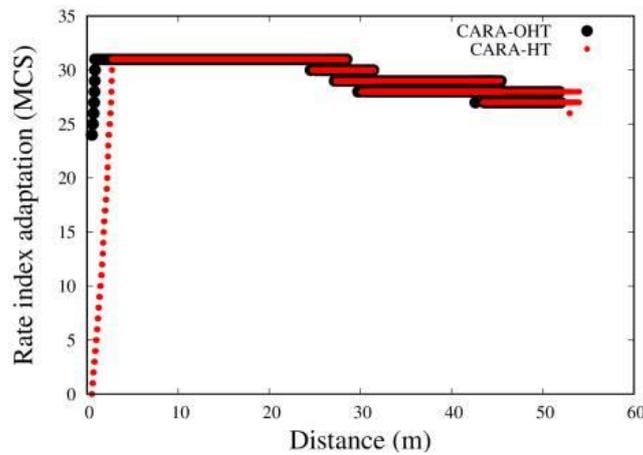


**Fig. 7.** Throughput comparison of the CARA-OHT, CARA-HT and ConstantRate algorithms based on channel conditions (without AMPDU scheme, 20 MHz channel bandwidth, GI=800 ns and 4 spatial streams)

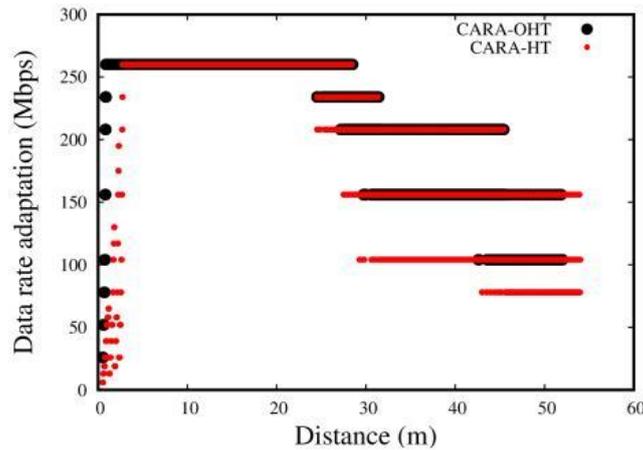
Figure 8 shows the adaptation of the MCS index of the CARA-OHT algorithm to changes in channel conditions and transmitted packets aggregated into the A-MPDU frame. The simulation results show that, at the beginning of transmission, the initial value used by the CARA-OHT algorithm was MCS-24. This initial value is in accord with the optimization of the initial value index rate designed in Section 3. If every threshold value indicating transmission success was reached, CARA-OHT incremented the index rate until it reached the MCS-31 index. The CARA-HT algorithm was seen using the initial value of the MCS-0 index and then incrementing it to the MCS-31 index. When channel quality decreased, at a distance of 50 m, both algorithms were seen to decrement the index rate to MCS-27. Thus, optimization of the initial value of the MCS index in CARA-OHT was successfully carried out because CARA-OHT reached the highest index rate faster than CARA-HT did.

Figure 9 shows the adaptation of the CARA-OHT algorithm data rate to changes in channel conditions and transmitted packets aggregated into the A-MPDU frame. When the distance between the STA and the AP was still < 5 m (channel conditions still good), the CARA-OHT algorithm was seen using the initial data rate of 26 Mbps. This initial

value is in accordance with the design optimization of the initial value index rate designed in Section 3. If every threshold value indicating transmission success was achieved, CARA-OHT increased the data rate to reach the highest data rate (260 Mbps). The CARA-HT algorithm was seen using the initial data rate of 6.5 Mbps and then increasing or decreasing the data rate according to the MCS index. Thus, the optimization of the initial value of the data rate, and of the method for increasing the data rate in CARA-OHT, was successfully carried out. There was no decrease in the data rate when continuous channel quality was good, relative to the rate with CARA-HT.

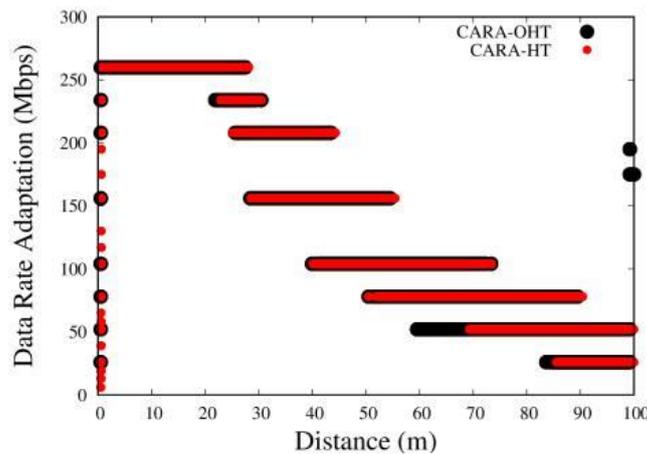


**Fig. 8.** Rate index adaptation of CARA-OHT and CARA-HT algorithms based on channel conditions (with AMPDU scheme, 20 MHz channel bandwidth, and 4 spatial streams)



**Fig. 9.** Data rate adaptation of CARA-OHT and CARA-HT algorithms based on channel conditions (with AMPDU scheme, channel bandwidth 20 MHz, and 4 spatial streams)

Figure 10 shows the CARA-OHT algorithm throughput for changing channel conditions and the transmitted packets aggregated into the A-MPDU frame. From the figure it can be seen that CARA-OHT throughput was better than the CARA-HT throughput, especially at the beginning of transmission (area X) and when there was a change in channel conditions from good to bad (area Y). When there was no collision in the network, CARA-OHT successfully adapted the data rate according to channel conditions so that the resulting throughput was higher up to 23 Mbps than the throughput generated by the CARA-HT algorithm. That proof that the CARA-OHT throughput was better is that the CARA-OHT throughput graph line is higher than the CARA-HT throughput line, especially in the circled areas.



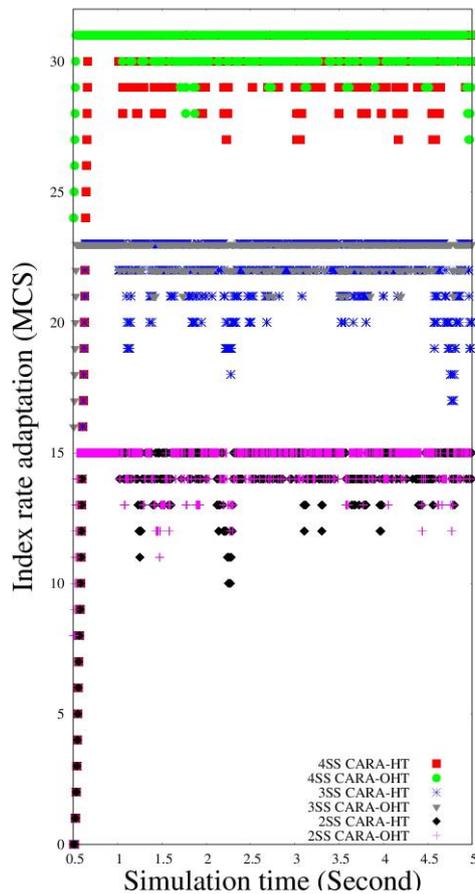
**Fig. 10.** Throughput of the CARA-HT and CARA-OHT algorithms based on channel conditions (with AMPDU scheme, 20 MHz channel bandwidth, SGI = 800 ns and 4 spatial streams)

## 5.2 CARA-OHT performance evaluation based on channel collision

The second evaluation was a simulation of the rate adaptation of the CARA-OHT algorithm based on the collision conditions. It was carried out using the topology shown in Figure 3. In the second simulation, three different PHY parameter scenarios were used: MIMO with two spatial streams (2SS), three spatial streams (3SS), and four spatial streams (4SS). However, the three scenarios used two additional parameters: bandwidth of 20 MHz and guard intervals of 800 ns. Thus, whereas the PHY layer in the AP and STA in the first scenario supported data rates ranging from MCS-0 (6.5 Mbps) to MCS-15 (130 Mbps), the PHY layer in the second scenario supported data rates from MCS-0 (6.5 Mbps) to MCS-23 (195 Mbps). Moreover, the PHY layer in the third scenario supported data rates ranging from MCS-0 (6.5 Mbps) to MCS-31 (260 Mbps), as shown in Table 1 (GI column = 800 ns, bandwidth 20 MHz).

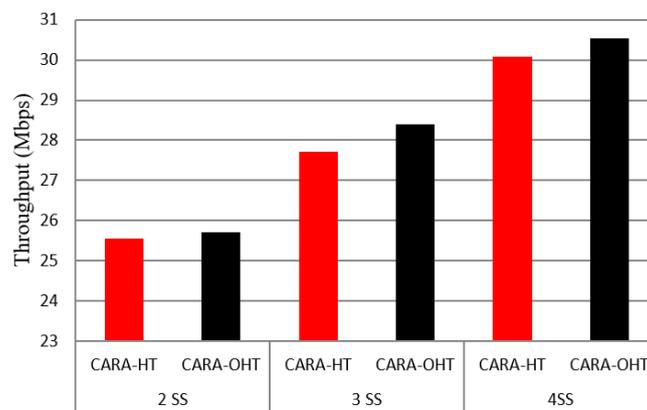
In the second simulation, the AP functioned as a recipient of the packets and ran the ConstantRate algorithm (apManager=ns3::ConstantRateWifiManager). Thus, each

packet received without error was confirmed by the AP by transmitting ACK using the highest constant data rate for this simulation scenario. Unlike in the first simulation, two STAs were used in the second simulation. The first STA was placed in a constant position 5 m to the right of the AP and was used to transmit UDP packets (1500 bytes) continuously to the AP using a data rate adapted by the CARA-OHT algorithm (staManager=ns3::CaraoHtWifiManager). The second STA was placed 5 m to the left of the AP and was enabled to transmit continuously UDP packets of 1500 bytes to the AP using a constant data rate adapted by (staManager=ns3::CaraoHtWifiManager). The data rate constant used was the highest data rate according to the second simulation scenario. Whereas in the first scenario (2SS) the constant data rate was MCS-15, in the second scenario (3SS), MCS-23 was the constant data rate, and in the third scenario (4SS) the constant data rate was MCS-31.



**Fig. 11.** Adaptation of the index rate of the CARA-OHT and CARA-HT algorithms based on collision channel (without AMPDU scheme, 20 MHz channel bandwidth, GI = 800 ns)

To create a collision condition, the channel model was changed to `RangePropagationLossModel` and the error model used was still `NistErrorRateModel`. Thus, the channel quality in the two STAs at a distance of 5 m from the AP became the same. Packets were transmitted in this second simulation scenario without using A-MPDU, and were transmitted for only 5 seconds, because it only evaluated whether changes in the adaptation rate function in CARA-OHT were working correctly. However, in simulations to evaluate throughput, packets were transmitted for longer periods (10 seconds) to ensure the average throughput produced. For comparison to the performance of the CARA-OHT algorithm, STA was also evaluated using the CARA-HT algorithm.



**Fig. 12.** Throughput of the CARA-OHT and CARA-HT algorithms based on collision channel (without AMPDU scheme, 20 MHz channel bandwidth, GI = 800ns)

Figure 11 shows the adaptation data rate of the CARA-OHT algorithm based on collision conditions. The evaluation results show, at the beginning of the simulation, the CARA-OHT algorithm successfully used the index rate according to the optimization design and coding implementation to optimize the initial index rate according to the PHY configuration used. The simulation results of the CARA-OHT algorithm show that, when used with 2SS, the initial value of the index rate was MCS-8, with 3SS the initial index rate was MCS-16, and with 4SS the initial index rate was MCS-24. In contrast, the CARA-HT algorithm always used the initial value of the MCS-0 rate index for PHY 2SS, 3SS, and 4SS configurations. When a collision occurred in a channel with 2SS, the CARA-OHT configuration only decreased the index rate to MCS-12, while CARA-HT decreased the index rate to MCS-10. When a collision occurred in a channel with a configuration of 3SS, CARA-OHT only decreased the index rate to MCS-21, while CARA-HT decreased the index rate to MCS-17. When a collision occurred in a channel with 4SS, the CARA-OHT configuration only decreased the index rate to MCS-28, while CARA-OHT decreased the index rate to MCS-27. The average use of a higher index rate in CARA-OHT when a collision occurred in a channel proves that the optimization design of the CARA-HT algorithm was successful.

Figure 12 shows the comparison of CARA-OHT and CARA-HT algorithm throughput based on collision conditions in the channel. The simulation results show that in the

scenarios with 2SS, 3SS, and 4SS, CARA-OHT throughput was higher 5~10% than the throughput produced by CARA-HT. This condition proves that the optimization of the CARA-OHT algorithm was able to use effectively the transmission mechanism in RTS/CTS mode if there was a collision on the channel. However, if there was no collision, CARA-OHT did not use the RTS/CTS mode. Thus, CARA-OHT can properly adjust the data rate and overhead of transmission, unlike CARA-HT, which cannot adapt the data rate or effectively use the RTS/CTS transmission mechanism when collisions occur in the channel.

### 5.3 CARA-OHT performance evaluation based on channel collision

Figure 13 shows the results of the CARA algorithm simulation in an IEEE 802.11s WMN according to the scenario described in Section 4. When the horizontal/vertical distance between the mesh-points was 21.21 m and the diagonal distance was 30 m, the constant data rate of 54 Mbps only produced 7.32 Mbps throughput, while the CARA algorithm produced throughput of 18.01 Mbps. Similarly, when the horizontal/vertical distance between the mesh-points was 69.29 m and the diagonal distance was 98 m, a constant data rate of 9 Mbps only produced 2.40 Mbps throughput. In this case, the CARA algorithm produced a throughput of 3.26 Mbps. All simulations show that the CARA algorithm throughput is higher 50~137% than that of the ConstantRate algorithm. Higher throughput is generated by the CARA algorithm because of the success of this algorithm in determining the best data rate for transmitting frames at a one-node mesh-point in an IEEE 802.11s WMN. While the ConstantRate algorithm that is used by default in an IEEE 802.11s WMN transmits frames through a longer route, causing lower throughput.

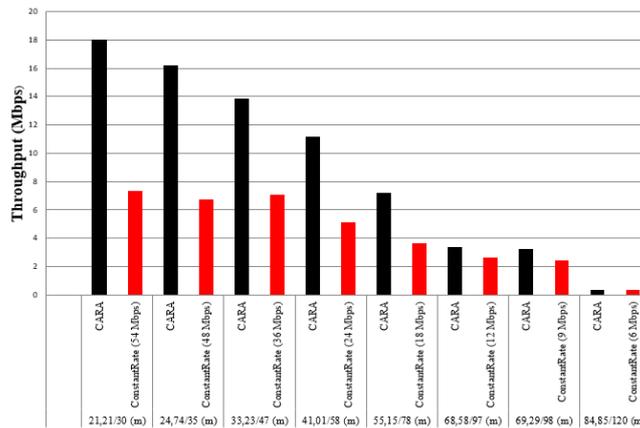


Fig. 13. Throughput comparison of the CARA and ConstantRate algorithms in IEEE 802.11s WMN

## 6 Conclusion

In the work reported in this paper, we optimized the CARA-HT algorithm in three parts: the initial index rate, the rate adaptation for channel quality improvement, and rate adaptation to channel quality degradation. Optimization of the initial index rate was done by determining the initial value of the index based on the HT-PHY configuration used. Optimization of the adaptation rate for improving channel quality was done by using a higher data rate only when the index rate was increased. In the reverse situation, the optimization of the rate adaptation to channel quality degradation was done by using a lower data rate only when the index rate was lowered. The simulation results show that the CARA-OHT algorithm produces higher throughput up to 23 Mbps at the start of transmission, and when there is an increase or decrease in the channel quality, compared to the CARA-HT algorithm with only 1 Mbps. The simulation results for collision conditions also show that the CARA-OHT algorithm produces higher throughput 5~10% than the CARA-HT algorithm. In simulation using nine nodes that functioned as mesh-points, all nodes were in constant position and the distance between mesh-points was the same horizontally and vertically. Each node could only communicate with the mesh-point next to it. The simulation results show that the CARA algorithm throughput is higher 50~137% than that of the constant rate algorithm. This proves that the CARA algorithm succeeded in determining the best data rate for frame transmission to the destination mesh-point node in an IEEE 802.11s WMN.

## 7 References

- [1] "IEEE Standard for Information Technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking," in IEEE Std 802.11s-2011, vol., no., pp.1-372, Sept. 10 2011. <https://doi.org/10.1109/ieeestd.2011.6018236>
- [2] "IEEE standard for information technology-- local and metropolitan area networks-- specific requirements-- part 11: Wireless LAN medium access control (MAC)and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput," IEEE Std 802.11n-2009, pp. 1–565, 2009. <https://doi.org/10.1109/ieeestd.2009.5307322>
- [3] "IEEE standard for information technology-- telecommunications and information exchange between systems local and metropolitan area networks-- specific requirements--part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications--amendment 4: Enhancements for very high throughput for operation in bands below 6 ghz." IEEE Std 802.11ac-2013, pp. 1–425, Dec 2013. <https://doi.org/10.1109/ieeestd.2017.7920364>
- [4] I. Pefkianakis, S. B. Lee and S. Lu, "Towards MIMO-Aware 802.11n Rate Adaptation," in IEEE/ACM Transactions on Networking, Vol. 21, No. 3, pp. 692-705, June 2013. <https://doi.org/10.1109/tnet.2012.2207908>
- [5] S. Seytnazarov and Young-Tak Kim, "Cognitive rate adaptation for high throughput IEEE 802.11n WLANs," 2013 15th Asia-Pacific Network Operations and Management Symposium (APNOMS), Hiroshima, Japan, pp. 1-6, 2013.

- [6] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, K. Almeroth, "A practical framework for 802.11 {MIMO} rate adaptation", *Computer Networks*, Vol. 83, No. pp. 332-348, 2015. <https://doi.org/10.1016/j.comnet.2015.03.015>
- [7] R. Karmakar, S. Chattopadhyay and S. Chakraborty, "Dynamic link adaptation for High Throughput wireless access networks," 2015 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Kolkata, pp. 1-6, 2015. <https://doi.org/10.1109/ants.2015.7413630>
- [8] R. Karmakar, S. Chattopadhyay and S. Chakraborty, "IEEE 802.11ac Link Adaptation under Mobility," 2017 IEEE 42nd Conference on Local Computer Networks (LCN), Singapore, pp. 392-400, 2017. <https://doi.org/10.1109/lcn.2017.90>
- [9] H. Munanzar, T. Y. Arif, Syahrial, " AARF-HT: Adaptive Auto Rate Fallback for High-Throughput IEEE 802.11n WLANs ", *International Journal of Communication Networks and Information Security (IJC-NIS)*, Vol. 10, No. 1, pp. 170-179, 2018.
- [10] T. A. Fridayansah, T. Y. Arif and R. Munadi, "Enhancement of Adaptive Auto Rate Fallback with Collision Detection for IEEE 802.11n/ac WLANs," 2018 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), Serpong, Indonesia, pp. 70-75, 2018. <https://doi.org/10.1109/icramet.2018.8683932>
- [11] T. Y. Arif, D. Rinaldi and R. Adriman, "Energy-Aware Rate Adaptation Algorithm for High Throughput IEEE 802.11n WLANs," 2018 3rd International Seminar on Sensors, Instrumentation, Measurement and Metrology (ISSIMM), Depok, Indonesia, pp. 48-53, 2018. <https://doi.org/10.1109/issimm.2018.8727738>
- [12] D. Passos and C.V.N. Albuquerque, "A Joint Approach to Routing Metrics and Rate Adaptation in Wireless Mesh Networks," *IEEE INFOCOM*, pp.1-2, 2009. <https://doi.org/10.1109/infcomw.2009.5072097>
- [13] B. Li, Y. Liu, K. Liu and Y. Liu, "QoS-Aware Routing with Rate Adaptation in Wireless Mesh Networks," 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, pp. 1-4, 2011. <https://doi.org/10.1109/wicom.2011.6040287>
- [14] TS Kim, G Jakllari, SV Krishnamurthy, M Faloutsos, "An integrated routing and rate adaptation framework for multi-rate multi-hop wireless networks," *Springer Wireless Networks*, Vol. 19, No. 5, pp. 985-1003, 2013. <https://doi.org/10.1007/s11276-012-0513-5>
- [15] M. R. Abid, S. Biaz, "Airtime ping-pong effect characterization in IEEE 802.11s wireless mesh networks", *Computing*, Vol. 97, No. 9, pp. 913-937, 2015. <https://doi.org/10.1007/s00607-013-0374-5>
- [16] S. Chakraborty and S. Nandi, "Data rate, path length and network contention trade-off in IEEE 802.11s mesh networks: A dynamic data rate selection approach," *Computer Networks*, Vol. 91, pp. 225–243, 2015. <https://doi.org/10.1016/j.comnet.2015.08.026>
- [17] W. J. Jung, J. Y. Lee, B. C. Kim, "Joint link scheduling and routing for load balancing in STDMA wireless mesh networks", *International Journal of Communication Networks and Information Security (IJC-NIS)*, Vol. 6, No. 3, pp. 246-252, 2014.
- [18] F. A. Setia, T. Y. Arif and R. Munadi, "Collision-Aware Rate Adaptation Algorithm for High-Throughput IEEE 802.11n WLANs," 2018 6th International Conference on Information and Communication Technology (ICoICT), Bandung, pp. 12-17, 2018. <https://doi.org/10.1109/icoict.2018.8528814>
- [19] A. Jönsson, D. Åkerman, E. Fitzgerald, C. Nyberg, B. E. Priyanto and K. Agardh, "Modeling, implementation and evaluation of IEEE 802.11ac in NS-3 for enterprise networks," 2016 Wireless Days (WD), Toulouse, pp. 1-6, 2016. <https://doi.org/10.1109/wd.2016.7461452>

- [20] K. Andreev, P. Boyko, "IEEE 802.11s Mesh Networking NS-3 Model", Workshop on ns-3 WNS3 2010 Malaga Spain, 2010.
- [21] T.Y. Arif, R. Munadi, Fardian, "Evaluation of the Minstrel-HT Rate Adaptation Algorithm in IEEE 802.11n WLANs", "International Journal of Simulation Systems, Science & Technology", Vol. 18, No. 1, pp. 11.1-11.7, March 2017.
- [22] N. S. Ravindranath, I. Singh, A. Prasad and V. S. Rao, "Performance Evaluation of IEEE 802.11ac and 802.11n using NS3", "Indian Journal of Science and Technology", Vol. 9, No. 26, pp. 1-9, July 2016.
- [23] A.B. Amewuda, F.A. Katsriku, and J.D. Abdulai, "Implementation and Evaluation of WLAN 802.11ac for Residential Networks in NS-3", Journal of Computer Networks and Communications, Vol. 2018, pp. 1-10, 2018. <https://doi.org/10.1155/2018/3518352>
- [24] M. Lavanya, C. S. Bindu, G. V. Kumar, "Internet Traffic based Channel Selection in Multi-Radio Multi-Channel Wireless Mesh Networks", International Journal of Communication Networks and Information Security (IJC-NIS), Vol. 11, No. 2, 2019. <https://doi.org/10.1109/issl.2017.8389234>
- [25] Alnabhan, M., Alshuqran, M., Hammad, M., and Nawayseh, M. A., "Performance evaluation of unicast routing protocols in manets - current state and future prospects", International Journal of Interactive Mobile Technologies, Vol. 11, No. 1, pp. 84 – 97, 2017. <https://doi.org/10.3991/ijim.v11i1.6295>
- [26] Allahham, A. A., Mohammed, M., Kadhim, N. S., "Multipath Routing Protocol Based on Cross-Layer Approach for MANET", International Journal of Interactive Mobile Technologies, Vol. 11, No. 1, pp. 71 – 83, 2017. <https://doi.org/10.3991/ijim.v11i1.6175>
- [27] Daoud, O., Hamarsheh, Q., Al-Sawalmeh, W., "MIMO-OFDM Systems Performance Enhancement Based Peaks Detection Algorithm", International Journal of Interactive Mobile Technologies, Vol. 7, No. 3, pp. 4 – 8, 2013. <https://doi.org/10.3991/ijim.v7i3.2302>

## 8 Authors

**Teuku Yuliar Arif** (corresponding author) is an associate professor in the field of computer and telecommunication network at Electrical and Computer Engineering Department, Faculty of Engineering, Universitas Syiah Kuala, Banda Aceh, Indonesia. His research interests include wireless and mobile networks, vehicular ad hoc networks (VANET), network security, network performance analysis and mesh network for Internet of Things (IoT). (e-mail: [yuliar@unsyiah.ac.id](mailto:yuliar@unsyiah.ac.id)).

**Munandar** is a Magister student in Electrical and Computer Engineering Department, Faculty of Engineering, Universitas Syiah Kuala, Banda Aceh, Indonesia. He is with the Research Group of Wireless and Mobile Network in Computer Network laboratory, Electrical and Computer Engineering Department, Faculty of Engineering, Universitas Syiah Kuala. (e-mail: [munandar@mhs.unsyiah.ac.id](mailto:munandar@mhs.unsyiah.ac.id)).

**Ramzi Adriman** is an assistance professor in the field of Computer Science and Information Engineering at Electrical and Computer Engineering Department, Faculty of Engineering, Universitas Syiah Kuala, Banda Aceh, Indonesia. He has research focuses on Optimization, Algorithm, Multi-Agent System, Game Theory, Network Security, High-Performance Computing, Parallel and Distributed Computing. (e-mail: [ramzi.adriman@unsyiah.ac.id](mailto:ramzi.adriman@unsyiah.ac.id)).

**Rizal Munadi** is an associate professor in the field of mobile and wireless communication at Electrical and Computer Engineering Department, Faculty of Engineering, Universitas Syiah Kuala, Banda Aceh, Indonesia. He is with the Research Group of Wireless and Mobile Network in Computer Network laboratory, Electrical and Computer Engineering Department, Faculty of Engineering, Universitas Syiah Kuala. (e-mail address: [rizal.munadi@unsyiah.ac.id](mailto:rizal.munadi@unsyiah.ac.id)).

Article submitted 2020-01-18. Resubmitted 2020-05-28. Final acceptance 2020-05-28. Final version published as submitted by the authors.