

# Parallel Rendezvous Protocol with Improved Broadcast Mechanism for Ad-Hoc Networks

**Mr. Joseph Kurian**

*Department of Computer Science and Engineering  
NCERC, University of Calicut Thrissur, Kerala, India*

**Mr. Siva Shankar. S**

*Department of Computer Science and Engineering  
NCERC, University of Calicut Thrissur, Kerala, India*

## Abstract

Wireless ad-hoc network technologies and standards such as IEEE 802.11s (WANET) are efficient, economic and simple to deploy. To increase the performance of ad hoc wireless networks, multiple rendezvous multi-channel MAC protocols are used. But this multi-channel MAC protocols has certain disadvantages such as broadcast issue, busy receiver problem, multi-channel hidden terminal problem etc. By using a proper broadcasting mechanism we can solve many of these problems. In this paper we are focusing on broadcast issue faced by some of the multiple rendezvous protocols and provide a solution for the problem. For this, we are introducing a special interface for broadcasting and a packet is broadcasted according to the node's behavior. Also we make a system study and comparison between the proposed work and multiple rendezvous protocol.

**Keywords:** Multi-channel, Multiple Rendezvous, Parallel Rendezvous, Independent hopping

## I. INTRODUCTION

Wireless technology has improved over the age due to the developing industry standards and deployment of wireless networking across wide commercial section. Wireless ad-hoc network technologies and standards, such as IEEE 802.11s (WANET) are cost-effective and easy to deploy, allow quick setup of a wireless network among a sequence of mobile stations, where the stations communicate with each other without the aid of an infrastructure (e.g., access points, cables or base stations). Because of the interference among nodes, the network performance of ad hoc networks is limited. Increasing the capacity of wireless networks to support multiple senders and receivers in wireless networks has been the focus of recent research [1]. Network capacity can be increased with the use of orthogonal channel approach.

Multiple devices can transmit in a parallel fashion on distinct orthogonal channels using a Multi-channel Media Access Control (MCMAC) protocol which in turn decreases the delay and increases the throughput [2][3][4]. In order to enhance the network performance, the MCMAC protocols switch between all of the available channels for ad hoc networks. Single Rendezvous (SR) and Multiple (parallel) Rendezvous (MR) protocols are two categories of protocols based on their operations on exchanging control information and data [2]. In Single Rendezvous, at any time a Single agreement is made between a pair of nodes on exchange of control information over a single channel. However in Multiple Rendezvous protocols, exchange of control information takes place between different node pairs over multiple channels at the same time. Thus, Multiple Rendezvous protocols can perform much better than Single Rendezvous protocols [2]. Issue of broadcast support reduces the efficiency of Multiple Rendezvous protocols. Reason is that the nodes might exist over different channels in Multiple Rendezvous multi-channel ad hoc networks and as a result some nodes might not receive broadcast information [1]. The hello packets used for the neighbors' discovery are also broadcasted in Multiple Rendezvous protocols which can be considered as another problem in Multiple Rendezvous protocols. Broadcasting is an important concept in the wireless ad-hoc networks. Majority of the problems such as network delay, multi-channel hidden terminal problem, network partitioning etc. that arises in the Multiple Rendezvous protocols can be addressed by the usage of a proper broadcasting method. So there is a need for optimal broadcasting method that requires deep research studies. This paper focuses on Multiple Rendezvous protocol's broadcast issue that causes degradation of the network performance. For this, we propose an additional interface especially for broadcasting, which will broadcast a Hello packet containing nodes' current condition. The broadcasting of packet is done according to change in the behavior of node. So neighboring node can easily understand the current status of that particular node at any time. Therefore so many problems that arise due to missing of broadcast packet and unknown condition of a node can be easily eliminated.

The remaining portion of the paper is organized as follows: Section 2 presents an overview of background and related works of this study. Section 3 discusses the Multiple Rendezvous protocols with Enhanced broadcast mechanism. A system study of the proposed Multiple Rendezvous technique with an analysis model is presented in section 4. Section 5 gives the simulation results of the proposed work and provides a comparison. Finally section 6 gives the conclusion.

## II. BACKGROUND AND RELATED WORKS

In this section, we will discuss about the widely deployed Multiple Rendezvous multi-channel MAC protocols and some of the important multi-channel issues affecting the parallel (multiple) rendezvous approach.

### A. Multiple Rendezvous Protocol

In Multiple Rendezvous multi-channel MAC protocols multiple independent channel hopping device pairs can make agreements simultaneously on different channels. The main objective is to overcome the single control channel bottleneck. Whenever there are Multiple Rendezvous channels, they need to be coordinated through a special mechanism so that two devices can meet on the same channel. Parallel Rendezvous or Multiple Rendezvous protocols such as McMAC and Slotted Seeded Channel Hopping (SSCH) with one radio and Dynamic Switching Protocol (DSP) with dual radio interface are mentioned here.

Like some of the Single Rendezvous protocols the time is divided into discrete slots or intervals in Multiple Rendezvous (MR) protocols [3]. Instead of sharing a common hopping sequence, nodes switch to (probably) different channels according to their individual sequence. Staying in a channel for a dwell time, nodes then iterate through their own hopping sequence. This is shown in Fig. 1. Then they overlap at least during one slot per sequence cycle, which permits them to exchange and learn each other's hopping sequence. When a sender wants to transmit data, it hops to the next channel of the receiver's hopping sequence. There are two variants at this point. In the first variant, the data transfer is performed over successive hops of the receiver's sequence, whereas in the second variant, the data transfer is completely performed in that channel. Slotted Seeded Channel Hopping (SSCH) is an example of the former variant [5] and DSP, McMAC [1][6] are examples of latter variant. By attaching its seed (used to compute the hopping sequence) into every packet each node announces its hopping sequence. Nodes then eventually learn each other's hopping sequences by seed broadcast mechanism.

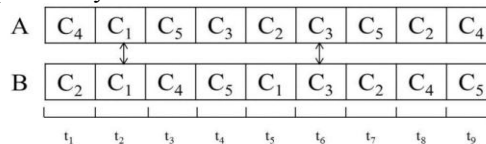


Fig. 1: Independent hopping of nodes

#### 1) Slotted Seeded Channel Hopping (SSCH)

SSCH is a well suited protocol for multi-hop wireless network. It is a distributed protocol and it does not require leader election or synchronization. A slot defined in the SSCH is the time spent on a single channel. The channel schedule consists of time at which node plans to make each switch and list of channels that the node plans to switch to in subsequent slots. A list of the channel schedules is maintained by all nodes for every other node it is aware of [5]. The channel schedule is represented as a set of pairs which is particularly denoted as  $(x_i, a_i)$  where  $x_i$  is current channel and  $a_i$  is seed that is a rule for updating the channel. The channel  $x_i$  and seed  $a_i$  are integer numbers and are represented in the range  $[0, 12]$  (13 possibilities) and  $[1, 12]$  respectively. Each node iterates through all of the channels in the current schedule, switching to the channel designated in the schedule in each new slot. The node then increments each of the channels in its schedule using the seed [5], and repeats the process. To preclude logical partitions one extra parity slot is inserted. After the node's iteration through every channel on each of its slots, it switches to a parity slot which is shown in Fig. 2. By occasionally broadcasting their channel schedule and offset inside this cycle nodes can realize one another's schedule [5].

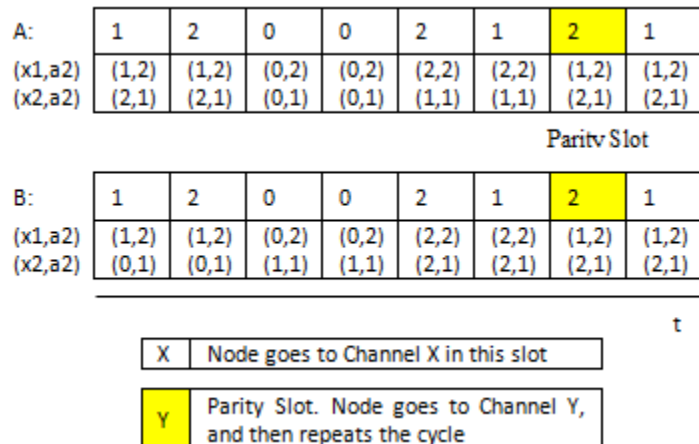


Fig. 2: Parity Slot Switching

#### 2) McMAC Protocol

In Multi-channel Mac conventions the transmitter must be aware of the channel on which its receiver is tuning. For this, McMAC utilize the MAC location of every node as a random number generator seed to create its hopping sequence known as the default sequence [6]. In a pseudo-random mode, every node hops over all M accessible channels. Time is partitioned into Big and Small Slots as demonstrated in Fig. 3. Every Small Slot ought to be generally equivalent to the short-term time it takes for the radio to recognize a bearer when utilizing Carrier Sense Multiple Access (CSMA). At Big Slot limits hopping happens. At each slot, the channel determined by the default sequence is the default channel of the node at that hop. At the point when

idle, a node listens on its default channel. Therefore, if a sender knows the seed, the hop limit, and the current balance into a recipient's sequence, it can foresee the hopping example of a receiver. Furthermore, the sender must know whether that channel is idle or not. McMAC utilize a blend of methods for this channel conditions. First and foremost, it utilizes conventional RTS/CTS system for IEEE 802.11 [7]. Then it listens for transporter in the wake of exchanging channel to dodge impact [6].

McMAC utilizes primary and backup system amid neighbor location. The whole hopping sequence of another neighbor can be precomputed and spared in memory. The rehashing sequence can be created utilizing a direct congruential generator [6]. In McMAC, each bundle sent incorporates the hopping signature of the sender, specifically the current (Time Slots) and the seed of the sender. The McMAC use synchronization calculations for overhauling the receiver's estimation of the clock float in respect to the sender. So it can precisely anticipate the default channel of the sender [6].

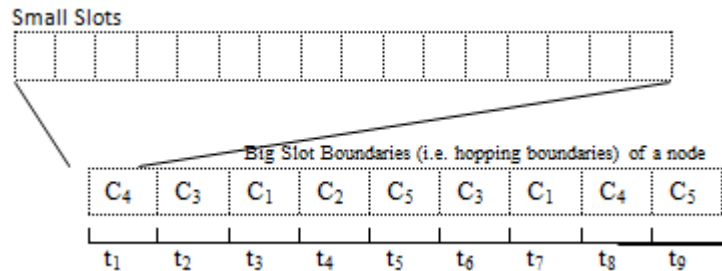


Fig. 3: Random Hopping Sequence of McMAC

### 3) Dynamic Switching Protocol (DSP) with dual radio

In DSP, node takes independent channel hopping and two interfaces that can switch alertly. One interface takes after fast hopping while the other one takes after slow hopping. Nodes ought to just synchronize with the slow hopping interfaces of one another utilizing pairwise synchronization and is universally synchronized. Hopping Control is utilized to produce hopping sequence for both the slow and fast hopping interfaces, as demonstrated in Fig. 4. Also, it promises that the two interfaces don't have the same channel in the meantime. The hopping sequence of the slow interface of a node is focused around a pseudo-arbitrary generator [1]. For the fast hopping interface, the hopping control produces a deterministic hopping sequence [1]. Nodes just need to know one another's hopping sequence of the slow hopping interface. New and existing nodes ought to intermittently transmit hello packets over channels. At the point when a node has a packet, the node figures out which channel the slow hopping interface of an end node is on. At that point, the node sends the packet through one of its two interfaces. The fast hopping interface is permitted to alter from current hopping sequence. The broadcast packet is transmitted over both interfaces [1]. For that two duplicates of the broadcast packet are created and transmitted.

	t <sub>1</sub>		t <sub>2</sub>		t <sub>3</sub>		
Channel 1	A <sub>1</sub>	C <sub>1</sub>	C <sub>1</sub>	D <sub>1</sub>	D <sub>2</sub>	B <sub>2</sub>	C <sub>2</sub>
Channel 2	B <sub>1</sub>	A <sub>2</sub>	D <sub>2</sub>	C <sub>2</sub>			C <sub>1</sub>
Channel 3		C <sub>2</sub>		A <sub>2</sub>	B <sub>2</sub>		A <sub>2</sub>
Channel 4	D <sub>1</sub>	B <sub>2</sub>	A <sub>1</sub>	B <sub>1</sub>	D <sub>2</sub>	A <sub>1</sub>	B <sub>1</sub>

A<sub>1</sub>, B<sub>1</sub>, C<sub>1</sub>, D<sub>1</sub> - Slow Hopping Sequence

A<sub>2</sub>, B<sub>2</sub>, C<sub>2</sub>, D<sub>2</sub> - Fast Hopping Sequence

Fig. 4: Hopping Sequence of DSP

### B. Multi-Channel Issues

There are several design considerations for multi-channel medium access control (MAC) protocol.

- 1) As some applications use broadcast information such as routing protocols it should support broadcast. Every nodes converse with each other over the same channel (if omni antennas are employed) in single-channel ad hoc networks and supports broadcast information. Nodes might exist over different channels in multi-channel ad hoc networks and due to that reason some nodes might not receive broadcast information. Higher delay or network partition will be incurred if broadcast support is not considered for designing MCMAC protocols [1].
- 2) Busy receiver problem is another issue that occurs only in multi-channel networks [2]. When the receivers are busy over other channels (either transmitting or receiving) transmitters cannot find their receivers on a channel where the receivers are supposed to be. This problem is termed as busy receiver problem which increases the dropping rate of packets and wastes the channel bandwidth.
- 3) Multi-channel hidden terminal problem is a remarkable problem that causes collisions [8]. To dispense with this problem, (RTS) request-to-send and (CTS) clear-to-send handshaking is utilized which cannot eliminate the Multi-channel hidden terminal problem totally. At the point when a transmitter has a packet for a recipient and the receiver is

on an alternate channel, the transmitter changes to the receiver's channel. Before sending a packet, the transmitter must locate the channel. The transmitter accepts that the channel is not moving in light of the fact that the transmitter is inside the transmission range (TR) of the recipient yet not inside the carrier sensing range (CSR) of the node that is at present transmitting to the receiver. At that point, the transmitter sends its packet to the receiver, and accordingly, a collision happens at the receiver which leads to degradation of network performance.

- 4) In single-channel networks, the single-channel exposed terminal problem is a customary issue, and there is no current answer for resolving it. This problem is not as genuine as the hidden terminal problem in light of the fact that the exposed terminal problem does not result in collisions; the single channel exposed terminal problem prompts poor channel use. In multi-channel networks, there is another sort of the exposed terminal problem known as the multi-channel exposed terminal problem because of poor channel assignment, which has not been decently mulled over.

### III. ENHANCED BROADCAST MECHANISM

In multiple rendezvous multi-channel MAC protocols, all nodes are mostly in different channels and there is no common channel for broadcasting. As a result the broadcasted packets may not be reached by all neighbors within the range which affects the performance of network. Also hello packets broadcasted are missed in the same manner and causes improper network updation which results in collision. The broadcast packet contains hello messages and routing information. Most of the routing protocol relies on broadcasting for routing information cause an issue in multiple rendezvous protocols. In this paper we focus on the broadcasting problem faced by the multiple rendezvous multi-channel MAC protocols. Also we are providing a solution to reduce collision caused by unknown condition of the neighbor nodes.

In our work, above problems are addressed by giving a dedicated interface with a common channel for broadcasting. And that channel is not taken as control channel or data channel. The multiple rendezvous strategy is not changed, communication taken place as in multiple rendezvous MAC protocol. Rendezvous taken place through default hopping channel and transreceiving occurs through the agreed upon data channel. We are taking a dedicated interface with a common broadcasting channel for all nodes. The nodes send their broadcast packet through this dedicated interface, which is reached by all neighbors within its range. The broadcasting through this common channel is done when nodes change its current state or need to make a routing calculation. So the network is updated when nodes change the state or topology, which provides the solution for the collision. The node changes its state when a communication takes place and may change its default hopping sequence for that communication. This may be unknown to its neighbor, which is already busy with another transreceiving. In our proposed method, node broadcast hello message when it changes the current state. This hello message reaches through dedicated interface of neighbors. Thus neighbors update the present condition of the node even in busy period.

The hello message contains an additional field to include the state of node. Already it contains the signature of the node which includes seed of the default hopping sequence, current time stamp and boundary limit for realign the slot. In addition to this we are including an extra field, current state of node. This field is also included in the nodes' neighbor records and routing table. A node updates the records and routing table as soon as it receives a broadcast packet and for communication it looks into the records and routing table. If its neighbor is using the same hopping channel and is busy, the node can easily understand that channel is in communication with its neighbor. Thus we can avoid collision due to unknown condition of the neighbor nodes.

#### A. Routing Protocol

We have to broadcast the packets dependably and adaptably. For this we adopted weighted cumulative expected transmission time protocol which is similar to Ad hoc On Demand Distance Vector (AODV) protocol. Another reason for choosing this protocol is that, it is best suited for Multi-Radio Multi-Hop wireless networks. It reduces intraflow interference which means reducing the number of nodes using same channel and allocates better available channels. To select the best path between a pair of source and destination nodes, on-demand weighted cumulative expected transmission time (WCETT) metric based routing [9] is adopted. The discovery of a route between source and destination nodes begins with source sending a route request packet across the network. The route request packet transmitted by a node on channel will be received by its neighbor nodes. The neighbor nodes calculate value of weighted cumulative transmission time and then forward. When a route request packet is received at a node, it starts resending the route request based on two conditions: In case the sequence number of route request packet is new, the weighted cumulative expected transmission time value of the path is stored in a routing table. Otherwise the sequence number of route request is repeated, that means a route request of the same sequence number has been processed. But its weighted value is smaller than the previous route request of the same sequence number. A destination node that receives the route request checks whether the received request's cost is smaller than the previously received cost with the same sequence number. If yes it sends back a route reply. Finally the source node will take the path having the lowest cost for data transmission and stores locally the other best paths.

### IV. SYSTEM STUDY

In this section, we briefly describe a system study of the proposed protocol with help of Bianchi's model [10]. We have adopted Bianchi's model [10] to analyze the throughput of our protocol because we are not changing the legacy IEEE 802.11 MAC protocols. At each and every time instance, the nodes randomly select the channels for subsequent transmissions. This is similar to the concept of throwing  $n$  balls (nodes) into  $k$  bins (channels). So, the number of nodes on a particular channel follows the

binomial distribution [10]. At all times, all nodes have packets for transmission (saturation condition) following Bianchi's approach, which means that each node will have a packet to transmit after every successful transmission. From [10], the transmission probability  $\tau$ , with which a node transmits packet over a channel, is given by

$$\tau = \frac{2(1 - 2p_c)}{(1 - 2p_c)(CW_{\min} + 1) + p_c CW_{\min}(1 - (2p_c)^m)} \quad (1)$$

Where  $p_c$  is the conditional collision probability seen by one node that transmitted its packet over one channel.  $CW_{\min}$  is defined as the minimum contention window size, and  $m$  denotes the maximum back-off stage. The probability  $p_c$  is defined as one or more of remaining nodes transmit their packets given that one node has already transmitted its packet on the same channel so that collision occurs over that particular channel. The probability  $p_c$  is assumed as independent and constant. This can be computed as follows:

$$\begin{aligned} p_c &= \sum_{i=1}^{n-1} (1 - (1 - \tau)^i) \binom{n-1}{i} \left(\frac{1}{k}\right)^i \left(1 - \frac{1}{k}\right)^{n-1-i} \\ &= 1 \left(1 - \frac{\tau}{k}\right)^{n-1} \end{aligned} \quad (2)$$

Note that the transmission probability  $\tau$  defined in Equation (1) is different from the transmission probability defined in Bianchi's paper as the conditional collision probability is different [10].  $\tau$  depends on the unknown variable  $p_c$  and Equations (1) and (2) can be solved numerically.

$P_i$  is the probability that there is no transmission (idle) in any given time over a particular channel and is given by:

$$\begin{aligned} P_i &= \sum_{j=0}^n ((1 - \tau)^j) \binom{n}{j} \left(\frac{1}{k}\right)^j \left(1 - \frac{1}{k}\right)^{n-j} \\ &= \left(1 - \frac{\tau}{k}\right)^n \end{aligned} \quad (3)$$

Let  $P_s$  be the probability of successful transmission over a particular channel provided exactly one station transmits over that channel (i.e., at least one node transmits).  $P_s$  is estimated as follows:

$$\begin{aligned} P_s &= \frac{\sum_{j=1}^n \binom{n}{j} \tau (1 - \tau)^{j-1} \left(\frac{1}{k}\right)^j \left(1 - \frac{1}{k}\right)^{n-j}}{1 - P_i} \\ &= \frac{\frac{n\tau \left(1 - \frac{\tau}{k}\right)^{n-1}}{1 - P_i}}{1 - P_i} \end{aligned} \quad (4)$$

The throughput  $\psi_l$  for channel  $l$  can be expressed as

$$\psi_l = \frac{P_s (1 - P_i) E[P]}{P_i \sigma + P_s (1 - P_i) T_s + (1 - P_s) (1 - P_i) T_c} \quad (5)$$

where  $E[P]$  denotes the average packet payload size,  $\sigma$  is the slot time,  $T_s$  is defined as average successful time because one node transmits over channel  $l$  successfully, and  $T_c$  is the average collision time in which channel  $l$  is sensed as being busy as two or more nodes transmit their packets results in collision. Total throughput for all channels is given by the equation

$$\Psi = \sum_{l=1}^k (\psi_l) \quad (6)$$

## V. PERFORMANCE EVALUATION

In this section we evaluate performance of the Enhanced broadcast mechanism and dynamic switching protocol (DSP) by multiple random topologies. The simulation results are used to compare the performance of proposed protocol with DSP in multi-hop networks. Here we have selected Dynamic switching protocol (DSP) for comparison as it out performs any other multiple rendezvous protocols. Even though both proposed work and DSP follows the same parallel rendezvous approach, Enhanced broadcast mechanism employs a dedicated channel for broadcasting whereas DSP uses multiple channels for broadcasting where collision or routing problems will be occurring due to broadcast packet missing.

The simulation results were compared with analysis model i.e. Bianchi's model and it can be seen that they are well matched. For analyzing the results a simulation environment (distribution of different channels over different nodes on the network) was setup where the collision probability is set high (high probability of collision occurrence in network).

### A. Simulation Settings

The nodes were placed in 1000x1000  $m^2$  and use IEEE.802.11 with FTP as traffic source. Each simulation is run for 960 seconds with increasing the number of nodes to get an accurate result. The simulation is done using CRCN patch into network simulator ns-2.31[11]. Simulation parameters are summarized in the following table 1:

Table – 1  
Simulation Parameters

Topology Area	1000 x 1000 m <sup>2</sup>
Mac Layer	IEEE 802.11
Transport Layer	TCP
Traffic Type	FTP
Packet Size	1024 Bytes
CW <sub>min</sub>	32
Queue Type	DropTail
Queue Length	50
Transmitter power	0.024 W
Receiver Power	0.036 W
Simulation Duration	960 Seconds

**B. Simulation Results**

We use the simulation results obtained from scenarios in which different nodes are kept in different channels which lead to missing of broadcast packet. On the basis of obtained results, we plot a graph between throughput of DSP, IEEE 802.11 and proposed protocol in multi-hop networks and it is as shown in Fig. 5.

From the graph shown below it is clear that our proposed work performs better than Dynamic switch Protocol and IEEE 802.11. The throughput obtained while using 3 channels as well as 6 channels is taken for comparing and analyzing the performance result. In DSP, as the no: of nodes increases the chance of missing broadcast packets increases which in turn affects the network performance and we may not get desired throughput. But in Enhanced broadcasting mechanism adopted multiple rendezvous protocol throughput is increased because the hello and routing packets are broadcasted in a more reliable manner.

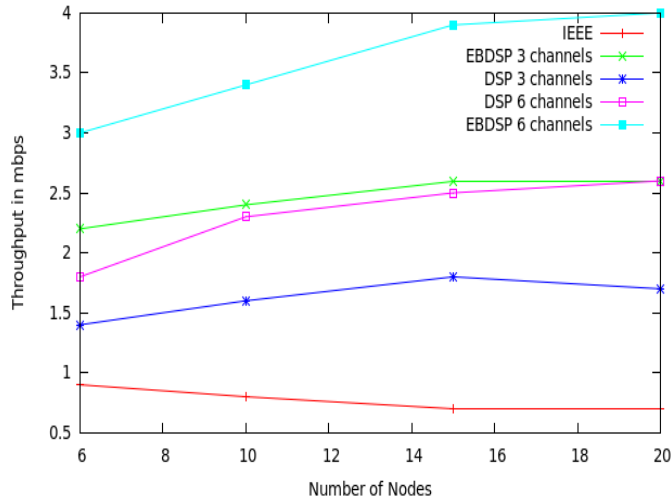


Fig. 5: Throughput vs No of Nodes

In Fig. 6, we plot the packet delivery ratio (i.e. packets that are successfully delivered to a destination with respect to the number of packets that have been generated by the sender) of DSP and EBDSP.

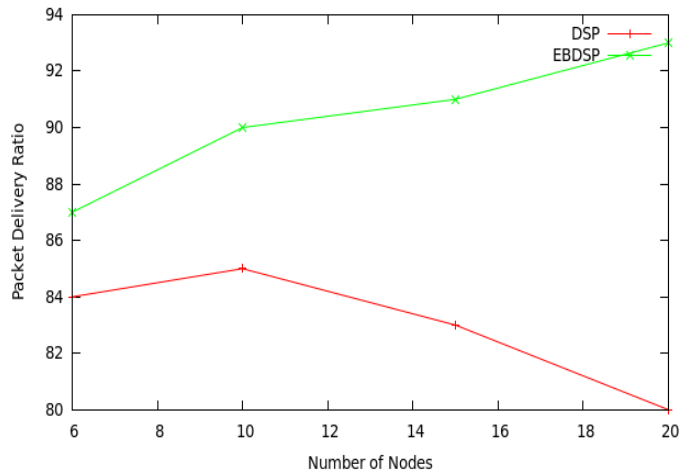


Fig. 6: Packet Delivery Ratio vs No of Nodes

In Fig. 6, it can be observed that the packet delivery ratio of Enhanced broadcast mechanism based multiple rendezvous protocol is higher than that of the DSP. The routing and channel selection must be proper for reliable delivery of packets in multi-channel MAC protocols. In DSP the routing and hello packets are broadcasted through multiple channels which affect the routing and communication process, but enhanced broadcast mechanism provides a better way of broadcasting which leads to higher packet delivery ratio.

Fig. 7 shows the average end-to-end packet delay (the time duration for a packet to be received correctly by its destination) which is important for real time applications. The factors affecting end to end delay are queuing, back-off, propagation, access, switching, and transmission time. The queuing size of each node is 50 packets. The proposed work achieves less delay compared to DSP because the channel selection and transmission are done at right time.

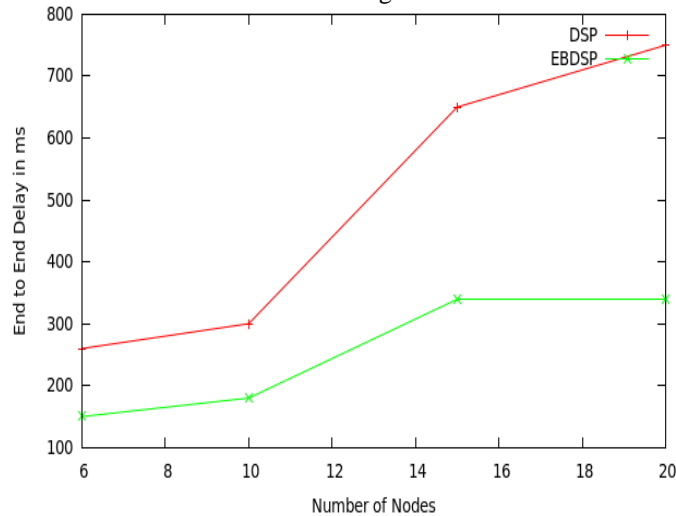


Fig. 7: Average End to End Delay vs No of Nodes

In Fig. 8, we present the normalized routing overhead of the proposed protocol and dynamic switching protocol. The normalized routing overhead is the total transmitted routing packets normalized by the total received packets. As our proposed protocol transmits broadcast packet through a dedicated interface and broadcast when there is a change in node's behavior, this approach minimize the overhead of discovering neighboring nodes and determines shorter routing paths. From Fig. 8, the routing overhead of our proposed protocol is very low when compared with that of dynamic switching protocol. This is due to the better route selection strategy as well as better channel selection.

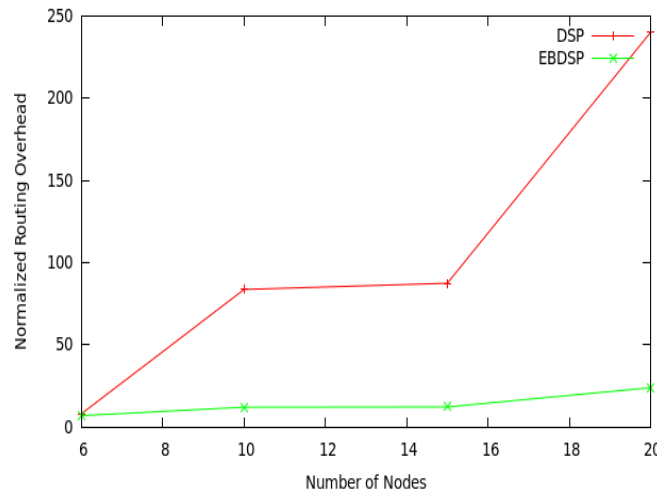


Fig. 8: Normalized Routing Overhead vs No of Nodes

## VI. CONCLUSION AND FUTURE SCOPE

The use of Multiple Rendezvous in multi-channel MAC protocols increases the performance of wireless ad-hoc networks than any other multi-channel protocols. But certain issues such as absence of routing information and network updation due to missing of broadcast packets affects the performance of Multiple Rendezvous protocols. This paper provides the solution for broadcast support through a dedicated interface at right time for broadcasting. At the same time, the proposed protocol broadcast hello packets whenever it is needed for network updation. A system study has been given to evaluate the network throughput. Simulation results have been provided to validate the analytical model and to demonstrate the improvement in the performance

of the network. The simulation results of our proposed work shows that, Multiple Rendezvous protocol with Enhanced Broadcast Mechanism outperforms any other Multiple Rendezvous multi-channel MAC protocols. The energy consumption issue due to the usage of multiple radios will be addressed in our future work.

## REFERENCES

- [1] Khaled H. Almotairi, Member, IEEE, and Xuemin (Sherman) Shen, Fellow, IEEE, "A Distributed Multi-Channel MAC Protocol for Ad Hoc Wireless Networks", IEEE Transactions on Mobile 1536-1233 (c) 2014 IEEE.
- [2] J. Mo, H.-S. So, and J. Walrand, "Comparison of multichannel MAC protocols," IEEE Trans. on Mobile Computing, vol. 7, no. 1, pp. 50–65, Jan. 2008.
- [3] J. Crichigno, M.-Y. Wu, and W. Shu, "Protocols and architectures for channel assignment in wireless mesh networks," Ad Hoc Networks, vol. 6, no. 7, pp. 1051 – 1077, 2008.
- [4] K. H. Almotairi and X. Shen, "Multichannel medium access control for ad hoc wireless networks," Wireless Communications and Mobile Computing (Wiley), vol. 13, no. 11, pp. 1047–1059, 2013.
- [5] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in Proc. of ACM MobiCom, 2004, pp. 216–230.
- [6] H.-S. W. So, J. Walrand, and J. Mo, "McMAC: A parallel rendezvous multi-channel MAC protocol," in Proc. Of IEEE Wireless Communications and Networking Conference (WCNC), Mar 2007, pp. 334–339.
- [7] IEEE standard 802.11, "Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) specifications," August 1999.
- [8] J. So and N. H. Vaidya, "Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," in Proc. of ACM MobiHoc, 2004, pp. 222–233.
- [9] R. Draves, J. Padhye, and B. Zill, "Routing in multiradio, multi-hop wireless mesh networks," in Int. Conf. on Mobile Computing and Networking (ACM Mobicom'04), 2004, pp. 114–128.
- [10] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," IEEE Journal on Selected Areas in Communications (JSAC), vol. 18, no. 3, pp. 535–547, March 2000.
- [11] "The Network Simulator NS-2," <http://www.isi.edu/nsnam/ns/>, 1996