RESEARCH ARTICLE OPEN ACCESS

# Self Reconfigurable Wireless Networks With Dsdv Protocol Implementation

K. Muthulakshmi* B. Balaji **, S. Gowtham**, M. Hemashri**, A. Jahana Bharvine**
*Associate professor, Department of Electronics & Communication Engineering
* * UG students, Department of Electronics & Communication Engineering,
Park College of Engineering and Technology, Coimbatore, Tamilnadu

**ABSTARCT:**
In multi hop wireless networks experience frequent link failures caused by channel interference, dynamic obstacles, and/or applications' bandwidth demands. These failures cause severe performance degradation in wireless networks or require expensive manual network management for their real-time recovery. This paper presents an autonomous network reconfiguration system (ARS) with destination sequence distance vector (DSDV) protocol that enables a multi radio Wireless network to autonomously recover from local link failures to preserve network performance. By using channel and radio diversities in Wireless networks, ARS generates necessary changes in local radio and channel assignments in order to recover from failures. Next, based on the thus-generated configuration changes, the system cooperatively reconfigures network settings among local mesh router. In this concept during the data transmission if the link fails in between the nodes, the previous node act as the header node. The header node, creating the loop around the neighboring nodes and find the energy efficient path, after finding the path send the data's towards it to reach the destination. Because of this there is no chance for data losing, Here ARS has been implemented and evaluated extensively on through ns2-based simulation. Our evaluation results show that ARS outperforms existing failure recovery schemes in improving channel-efficiency .

**Index Terms** — DSDV protocol, IEEE 802.11, multi radio wireless networks , self-reconfigurable networks, wireless link failures,

## I. INTRODUCTION:

Wireless networks are being developed actively and deployed widely for a variety of applications, such as public safety, environment monitoring, and citywide wireless Internet services [1] –[3]. They have also been evolving in diverse patterns (e.g., using multi radio/channel systems [4] –[7]), however, due to heterogeneous and fluctuating wireless link conditions [8] –[10], preserving the required performance of such wireless Networks is still a challenging problem. For example, some links of a Network may experience significant channel interference from other coexisting wireless networks. Some parts of networks might not be

Able to meet increasing bandwidth demands from new mobile users and applications. Links in a certain area (e .g.., a hospital) might not be able to use some frequency channel because of spectrums etiquette or regulation[11].

Able to meet increasing bandwidth demands from new mobile users and applications. Links in a certain area (e.g.., a hospital) might not be able to use same frequency channel because of spectrums etiquette or regulation [11].

Even though many solutions for wireless networks to recover from wireless link failures have been proposed, they still have several limitations as follows. First, resource-allocation algorithms can provide [11] -[14] (theoretical) guidelines for initial network resource planning ,However even though their approach provides a comprehensive optimal network configuration plan they often require the Global" configuration changes, which are undesirable in the case of frequent local link failures. Next, a *greedy* channel-assignment algorithm [15] can reduce the requirement of network changes by changing the settings of only the faulty link. However, this greedy change might not be able to realize full improvements, which can only be achieved by considering configurations of neighboring mesh routers in addition to the faulty link. Third, fault-tolerant routing protocols, such as local rerouting [16] -[17] or multipath routing, can be adopted to use network-level path diversity for avoiding the faulty links. However, they rely on de-tour paths or redundant transmissions, which may require more network resources than link-level network reconfiguration.

To overcome the above limitations, we propose an *autonomous network reconfiguration system* (ARS) that allows a multi radio Wireless network to autonomously reconfigure its local network settings—channel, radio, and route assignment for real-time recovery from link failures. In its core, ARS is equipped with a reconfiguration planning algorithm that identifies local configuration changes for the recovery while minimizing changes of healthy network settings. Briefly, ARS first searches for feasible local configuration changes available around a faulty area, based on current channel and radio associations. Then, by imposing current network settings as constraints, ARS identifies reconfiguration plans that require the minimum number of changes to the healthy network settings.

Next, ARS also includes a monitoring protocol that enables a wireless network to perform real-time failure recovery in conjunction with the planning algorithm. The accurate link-quality information from the monitoring protocol is used to identify network changes that satisfy applications' new QoS demands or that avoid propagation of QoS failures to neighboring links (or "ripple effects"). Running on every mesh node, the monitoring protocol periodically measures wireless link conditions via a hybrid link-quality measurement technique. Based on the measurement information, ARS detects link failures and/or generates [18] QoS-aware network recon-figuration plans upon detection of a link failure. ARS have been implemented and evaluated extensively via experimentation on our multi radio, wireless network test-bed as well as via ns2-based simulation. Our evaluation results show that ARS outperforms existing failure-recovery methods such as static or greedy channel assignment and local re-routing. First, ARS's planning algorithm effectively identifies reconfiguration plans that maximally satisfy the applications' QoS demands, accommodating twice more flows than static assignment. Next, ARS avoids the ripple effect via QoS-aware reconfiguration planning, unlike the greedy approach. Third, ARS's local reconfiguration improves network throughput and channel efficiency by more than 26% and 92%, respectively, over the local rerouting scheme.

## II. EXISTING SYSTEM:

Even though many solutions for WMNs to recover from wireless link failures have been proposed, they still have several limitations as follows. First, resource-allocation algorithms can provide (theoretical) guidelines for initial network resource planning. However, even though their approach provides a comprehensive and optimal network configuration plan, they often require "global" configuration changes, which are undesirable in the case of frequent local link failures. Next, a *greedy* channel-assignment algorithm can reduce the requirement of network changes by changing settings of only the faulty link(s). However, this greedy change might not be able to realize full improvements, which can only be achieved by considering configurations of neighboring mesh routers in addition to the faulty link(s). Third, fault-tolerant routing protocols, such as local rerouting or multipath routing, can be adopted to use network-level path diversity for avoiding the faulty links. However, they rely on detour paths or redundant transmissions, which may require more network resources than link-level network reconfiguration.

### 2.1 METHODOLOGY USED:

2.1.1 ARS algorithm

2.1.2 Greedy channel assignment

2.1.3 QOS demands

2.1.4 IEEE 802.11

2.1.5 AODV protocol

### 2.2 Limitation of existing approaches:

Given the above system models, we now discuss the pros and cons of using existing approaches for self-reconfigurable wireless networks.

#### 2.2.1 Localized Reconfiguration*:*

Network reconfiguration needs a planning algorithm that keeps necessary network changes (to recover from link failures) as local as possible, as opposed to changing the entire network settings. Existing channel assignment and scheduling algorithms[12]-[14] provide holistic guidelines such as throughput bounds and schedule ability for channel assignment during a network deployment stage. However, the algorithms do not consider the degree of configuration changes from previous network settings, and hence they often require *global* network changes to meet all the constraints, akin to edge coloring problems. Even though these algorithms are suitable for static or periodic network planning, they may cause network service disruption, and thus are unsuitable for dynamic network reconfiguration that has to deal with frequent local link failures. Next, the *greedy* channel-assignment algorithm, [15] which considers only local areas in channel assignments, might do better in reducing the scope of network changes than the above-mentioned assignment algorithms. However, this approach still suffers from the ripple effect, in which one local change triggers the change of additional network settings at neighboring nodes due to association dependency among neighboring radios. This undesired effect might be avoided by transforming a mesh topology into a tree topology, but this transformation reduces network connectivity as well as path diversity among mesh nodes. Finally, interference-aware channel-assignment algorithms [5] can minimize interference by assigning orthogonal channels as closely as possible geographically. While this approach can improve overall network capacity by using additional channels, the algorithm could further improve its flexibility by considering both radio diversity (i.e., link association) and local traffic information. If channel 5 is lightly loaded in a faulty area, the second radio of node $C$ can re -associate itself with the first radio of node, avoiding configuration changes of other links.

**2.2.2 QoS-Awareness:** Reconfiguration has to satisfy QoS constraints on each link as much as possible. First, given each link's bandwidth constraints, existing channel-assignment and scheduling algorithms[5] [13] [15] can provide approximately optimal network configurations. However, as pointed out earlier, these algorithms may require global network con-figuration changes from changing local QoS demands, thus causing network disruptions. We need instead a reconfiguration algorithm that incurs only local changes while maximizing the chance of meeting the QoS demands. For example, if link experiences a QoS failure on channel 1, then one simple reconfiguration plan would be to re-associate R1 of node H to R2 of node E in channel 5, which has enough bandwidth.

**2.2.3Cross-Layer Interaction*:*** Network reconfiguration has to jointly consider network settings across multiple layers. At the network layer, fault-tolerant routing protocols, such as local rerouting [16] or multipath routing [17], allow for flow reconfiguration to meet the QoS constraints by exploiting path diversity. However, they consume more network resources than a link reconfiguration because of their reliance on detour paths or redundant transmissions. On the other hand, channel and link assignments across the network and link layers can avoid the overhead of detouring, but they have to take interference into account to avoid additional QoS failures of neighboring nodes.

## 2.3 REVIEW OF AODV PROTOCOL:

AODV is a single-path, on-demand routing protocol. When a source node, NS, generates a packet to a particular destination node, ND, it broadcasts a route request (RREQ) [22] packet. The RREQ contains the following fields [23]:

<Source IP address,
Source sequence number,
broadcast ID,
Destination IP address,
Destination sequence
number, hop-count>,

Where the source and destination IP addresses remain constant for the lifetime of the network, the source sequence number is a monotonically increasing indicator of pocket "freshness," destination sequence number is the last known sequence number for $n_d$ at $n_s$ and hop-count is initialized to zero and incremented at each intermediate node which processes the RREQ. A RREQ is uniquely identified by the combination of a source sequence number and broadcast ID. An intermediate node only processes a RREQ if it has not received a previous copy of it. If an intermediate node has a route to Nd with a destination sequence number at least that in the RREQ, it returns a route reply (RREP) packet, updated with the information that it has. If not, it records the following information: source IP address, source sequence number, broadcast ID, destination IP address and expiration time for reverse path route entry, and forwards the RREQ to its neighbors.

Like the RREQ, a RREP is only processed on first sighting and is discarded unless it has a greater destination sequence number than the previous RREP or the same destination sequence number but a smaller hop-count. The RREP packet contains the following fields:

<Source IP address,
Destination IP address,
Destination sequence
number, hop-count,
Route expiration time>.

## III. PROPOSED WORK:

To overcome the above limitations, we propose an *autonomous network reconfiguration system* (ARS) that allows a multi radio WMN (mr-WMN) to autonomously reconfigure its local network settings—channel, radio, and route assignment—for real-time recovery from link failures. In its core, ARS is equipped with a reconfiguration planning algorithm that identifies local configuration changes for the recovery while minimizing changes of healthy network settings. ARS also include a monitoring protocol that enables a WMN to perform real-time failure recovery in conjunction with the planning algorithm. The accurate link-quality information from the monitoring protocol is used to identify network changes that satisfy applications' new QoS demands or that avoid propagation of QoS failures to neighboring links (or "ripple effects"). Running on every mesh node, the monitoring protocol periodically measures wireless link conditions via a hybrid link-quality measurement technique, as we will explain in Section IV. Based on the measurement information, ARS detects link failures and/or generates QoS-aware network reconfiguration plans upon detection of a link failure.

### 3.1 ARS ARCHITECTURE

ARS reconfiguration planning algorithm that identifies local configuration changes for the recovery while minimizing changes of healthy network settings. Briefly, ARS first searches for feasible local configuration changes available around a faulty area, based on current channel and radio associations. Then, by imposing current network settings as constraints, ARS identifies reconfiguration plans that require the minimum number of changes to the healthy network settings. We first present the design rationale and overall algorithm of ARS. Then, we detail ARS's reconfiguration algorithms. Finally, we discuss the complexity of ARS.

### 3.2 . Overview

ARS is a distributed system that is easily deployable in IEEE 802.11-based Wireless networks. Running on every mesh node, ARS supports self reconfigurability via the following distinct features.

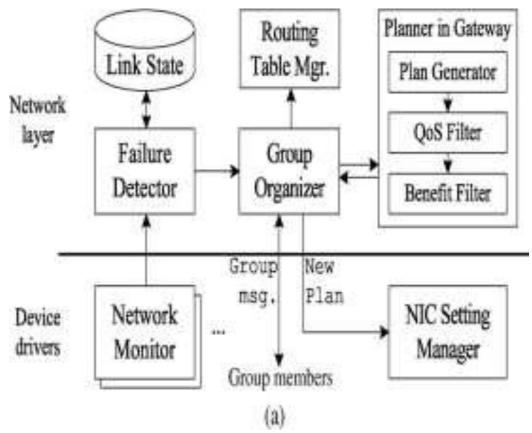


**Fig 3.1.1 (a) ARS architecture for the node**

**32.1 Localized reconfiguration:** Based on multiple channels and radio associations available, ARS generates re configuration plans that allow for changes of network configurations only in the vicinity where link failures occurred while retaining configurations in areas remote from failure locations.

**3.2.2 QoS-aware planning:** ARS effectively identifies QoS-satisfiable reconfiguration plans by: 1) estimating the QoS-satisfiability of generating reconfiguration plans; and 2) deriving their expected benefits in channel utilization.

**3.2.3 Autonomous reconfiguration via link-quality monitoring:** ARS accurately monitors the quality[4] of links of each node in a distributed manner. Furthermore, based on the measurements and given links' QoS constraints, ARS detects local link failures and autonomously initiates network reconfiguration.

**3.2.4 Cross-layer interaction:** ARS actively interact across the network and link layers for planning. This interaction enables ARS to include a rerouting for reconfiguration planning in addition to link-layer reconfiguration. ARS can also maintain connectivity during the recovery period with the help of a routing protocol.

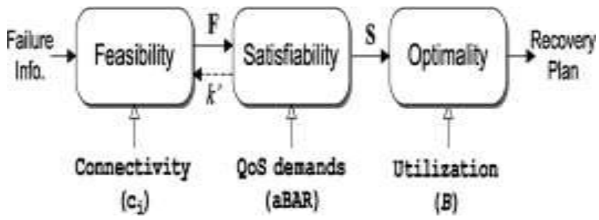


**Fig-3.1.2). ARS Reconfiguration Planning Method**

### 3.2.5 Planning for Localized Network Reconfiguration

The core function of ARS is to *systematically* generate localized reconfiguration plans. A *reconfiguration plan* is defined as a set of links' configuration changes (e.g., channel switch, link association) necessary for a network to recover from a link(s) failure on a channel, and there are usually multiple reconfiguration plans for each link failure. Existing channel-assignment and scheduling algorithms [5] [13] [15] seek "optimal" solutions by considering tight QoS constraints on all links, thus requiring a large configuration space to be searched and hence making the planning often an NP-complete problem [5]. In addition, change in a link's requirement may lead to completely different network configurations. By contrast, ARS systematically generates re-configuration plans that localize network changes by dividing the reconfiguration planning into three processes—feasibility, QoS satisfiability, and optimality—and applying different levels of constraints. As depicted in Fig. 3.1.2, ARS first applies connectivity constraints to generate a *set* of feasible reconfiguration plans that enumerate feasible channel, link, and route changes around the faulty areas, given connectivity and link-failure constraints. Then, within the set, ARS applies strict constraints (i.e., QoS and network utilization) to identify a reconfiguration plan that satisfies the QoS demands and that improves network utilization most.

### 3.3MODEL DESIGN OF PROPOSED WORK:

In this design work we consider the 6 nodes from that we can communicate or transfer the data packet to every node through routing table. Once the transmitter fixed to send the data to the desired node in between any problem may occur by the time the router select the best optimal path to transfer the nodes .Before that one node sends the hello packets to the remaining other nodes to check the channel available or not. After checking the channel availability fixes the source and destination node. Immediately data will be transfer if any problem between the sending and receiving nodes (i.e.) link failure .Previous node act as the header and the maintain the loop around the neighboring node finding the most energy efficient path to reach the destination. This is the main concepts of this project because of that no data loss will be occurring.

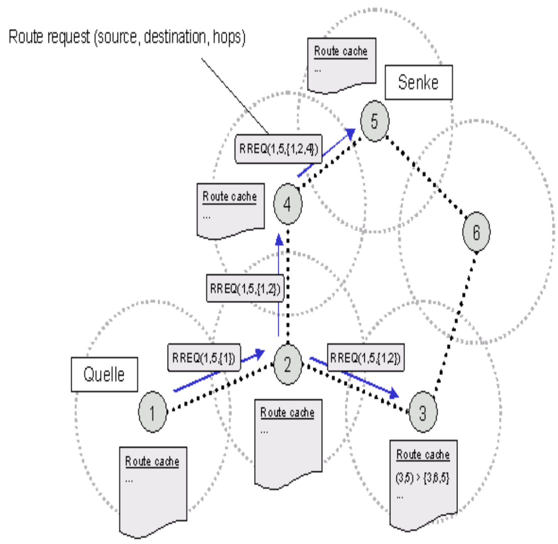


**Fig 3.3.1 model design of proposed work**

### 3.4 DSDV PROTOCOL:

Destination sequenced distance vector routing (DSDV) is adapted from the conventional Routing Information Protocol (RIP) to ad hoc network routing. It adds a new attribute, sequence number, to each route table entry of the conventional RIP. Using the newly added sequence number, the mobile nodes can distinguish stale route information from the new and thus prevent the formation of routing loops. **Packet Routing and Routing Table Management** [19] In DSDV, each mobile node of an ad hoc network maintains a routing table, which lists all available destinations, the metric and next hop to each destination and a sequence number generated by the destination node. Using such routing table stored in each mobile node, the packets are transmitted between the nodes of an ad hoc network. Each node of the ad hoc network updates the routing table with advertisement periodically or when significant new information is available to maintain the consistency of the routing table with the dynamically changing topology of the ad hoc network. Throughput is the number of packets that is passing through the channel in a particular unit of time. This performance metric shows the total number of packets that have been successfully
Delivered from source node to the destination node and it can be improved with increasing node density.

**Fig3.4.1** shows the sending throughput for UDP from the source node. It is observed that sending throughput, maximum in the time interval of 200 to 250 for both routing protocol and it is increased because of node density, less traffic and free channel. At the rest of the time sending throughput is almost constant. Here, AODV performance is better than DSDV in terms of sending throughput.
**Fig 3.4.2** shows the sending throughput for TCP packets. In the time interval of 200 to 250 maximum amount of TCP packets have been delivered from source to destination node in terms of DSDV because it is a proactive type routing protocol and advantage of these types of protocols is there are no delay to find out the route from source to destination nodes because the path is immediately available when source need to send a packet [2].

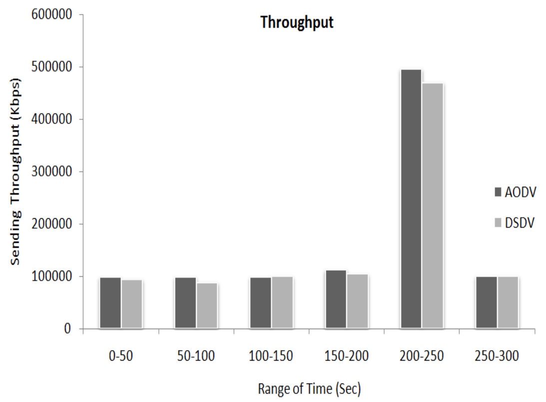


**Fig 3.4.1transmitter - throughput analysis between AODV and DSDV protocol in UDP**

**Fig3.4.3** shows the maximum receiving throughput in the time interval of 100 to 150 in terms of DSDV as well as a maximum amount UDP packets actually received by the particular destination because in that particular time interval the send node and receive node distance is less, free of channel for those packets. On the other side, AODV shows better performance and the performance rate sequentially increasing.

**Fig 3.4 4** shows the receiving throughput for TCP packets that is maximum in the time range of 100 to 150 for DSDV protocol because, its maintain a periodic table which broadcast routing table continuously to its neighbors for the update. For the same time interval AODV decrease because of less active route.

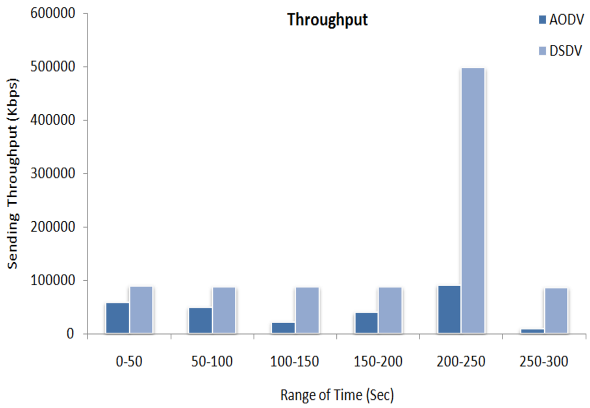


**Fig 3.4.2 transmitter of throughput analysis between AODV and DSDV in TCP**

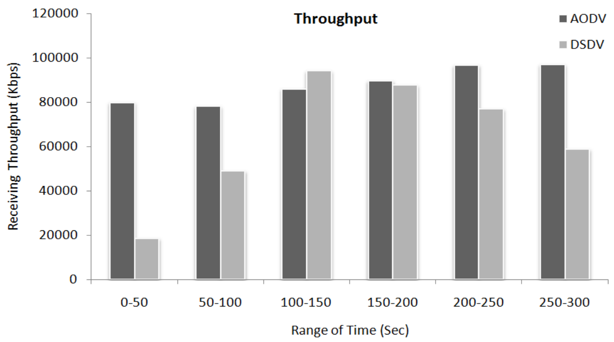


**Fig 3.4.3 receiving throughput analysis between AODV and DSDV in UDP**

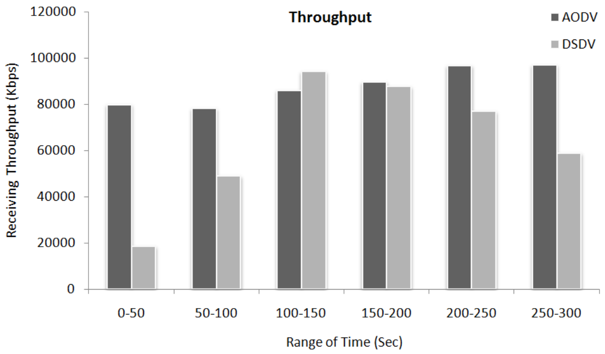


**Fig 3.4.4 receiving throughput analysis between AODV and DSDV in TCP**

Periodically or immediately when network topology changes are detected, each mobile node advertises routing information using broadcasting or multicasting a routing table update packet. The update packet starts out with a metric of one to direct connected nodes. This indicates that each receiving neighbor is one metric (hop) away from the node. It is different from that of the conventional routing algorithms. After receiving the update packet, the neighbors update their routing table with incrementing the metric by one and retransmit the update packet to the corresponding neighbors of each of them. The process will be repeated until all the nodes in the ad hoc network have received a copy of the update packet with a corresponding metric. The update data is also kept for a while to wait for the arrival of the best route for each particular destination node in each node before updating its routing table and retransmitting the update packet. If a node receives multiple update packets for a same destination during the waiting time period, the routes with most recent sequence numbers are always preferred as the basis for packet forwarding decisions, but the routing information is not necessarily advertised immediately, if only the sequence numbers have been changed. If the update packets have the same sequence number with the same node, the update packet with the smallest metric will be used and the existing route will be discarded or stored as a less preferable route. In this case, the update packet will be propagated with the sequence number to all mobile nodes in the ad hoc network. The advertisements of routes that are about to change may be delayed until the best routes have been found. Delaying the advertisement of the possibly unstable route can damp the fluctuations of the routing table and reduce the number of rebroadcasts of possible route entries that arrive with the same sequence number.

The elements in the routing table of each mobile node change dynamically to keep consistency with dynamically changing topology of an ad hoc network. To reach this consistency, the routing information advertisement must be frequent or quick enough to ensure that each mobile node can almost always locate all the other mobile nodes in the dynamic ad hoc network. Upon the updated routing information, each node has to relay data packet to other nodes upon request in the dynamically created ad hoc network.

### 3.5 DISADVANTAGES OF DSDV PROTOCOL:

The main purpose of DSDV is to address the looping problem of the conventional distance vector routing protocol and to make the distance vector routing more suitable for ad hoc network routing. However, DSDV arises route fluctuation because of its criteria of route updates. At the same time, DSDV does not solve the common problem of all distance vector routing protocols, the unidirectional links problem

## 3.6.SIMULATION TOOL DESCRIPTION:

SIMULATION TOOL (NETWORK SIMULATOR 2) Network Simulator (Version 2), widely known as NS2, is simply an event driven simulation tool that has proved useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2 It consists of two simulation tools. The network simulator (ns) contains all commonly used IP protocols. The network animator (nam) is use to visualize the simulations. Ns-2 [20] fully simulates a layered network from the physical radio transmission channel to high-level applications the simulator was originally developed by the University of California at Berkeley .

recently extended to provide simulation support for ad hoc network by Carnegie Mellon University (CMU Monarch Project homepage, 1999) NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl) while the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e., a frontend). As shown in figure The C++ and the OTcl are linked together using TclCL After simulation, NS2 outputs either text-based or animation-based simulation results. To interpret these results graphically and interactively, tools such as NAM (Network Animators) and XGraph are used.

## 3.7HARDWARE SPECIFICATION

To develop this system with IBM compatible personal computer with Pentium IV processor was used.

| | | |
|---|---|---|
| Main processor | : | Pentium IV processor 1.13 GHz |
| Hard disk capacity | : | 40GB |
| Cache memory | : | 512 MB |
| Monitor | : | LG Digital Color Monitor |
| Keyboard | : | Samsung |
| Mouse | : | Logitech |

## 3.8 SOFTWARE SPECIFICATION

| | | |
|---|---|---|
| Operating system | : | Fedora 8 (linux) |
| Scripting language | : | Network Simulator 2.33 |
| Protocol developed | : | C++ |
| Scripting | : | Tool Command Language |

## IV MODULES:

4.1 Constructing mesh topology network in the NS 2 and
Hello packet sending.
4.2 Routing path between Source and destination.
4.3 Implementation of ARS algorithm.

### 4.1. CONSTRUCTION OF MESH TOPOLOGY NETWORK IN NS2:

Initial step of the ns 2 simulator to create the mesh topology nodes in the ns 2 graph space .after that each node sending the hello packets to the other nodes to check whether the channel free or not .if the channel is free it can send the data transmission towards the channel else it will change the path for the data transmission to the available path. The following fig 5.1.1 shows the ns2 simulator outputs explaining about the construction of mesh topology .

### 4.2.ROUTING PATH BETWEEN SOURCE AND DESTINATION:

After the above mentioning process every node generates the dynamic route path for the data transmission. After that the user fixing the source and destination nodes in simulator coding the nodes are ready to send the data's. When the program is run the data will be continuously sent. The following fig 5.1.2 shows that routing path in the ns2 simulator.

### 4.3.IMPLEMENTATION OF ARS ALGORITHM:

In the ns2 simulator, thus the above process is common. That is, data transfer between the source and destination. During the data transferring link failure and other losses will be occur between the source and destination points. Because of that problem additional process is added to the mechanism that is autonomous reconfiguration system . 1) *network planner*, which generates reconfiguration plans only in a gateway node; 2) *group organizer*, which forms a local group among mesh routers; 3) *failure detector*, which periodically interacts with a network monitor in the device driver and maintains an up-to-date link-state table ;and 4) *routing table manager*, through which ARS obtains or updates states of a system routing table .the following steps which is used to implement the ARS[23] algorithm.

**Monitoring period**
1: For every link j do
2:Measure link quality using passive Monitoring
3: End for
4: Send monitoring results to gateway g
**Failure detection and group formation**

**Period**
5:if link violates link requirements then
6: request a group formed on channel of
Link ;
7: end if
8**:** participate in a leader election if a request is
Received
**Planning period**
9: if node is elected as a leader then
10: send a planning request message to a
Gateway;
11: else if node is a gateway then
12:synchronize requests from reconfiguration
Groups
13:generate a reconfiguration plan for ;
14: send a reconfiguration plan to a leader of
;
15: End if
**Reconfiguration period**
16: if includes changes of node then
17: apply the changes to links
18: end if
19: relay it to neighboring members, if any

During the period of data transferring time if the data loss occurs, the previous node act as the header node , create the loop around the neighboring nodes. Then the header nodes make the decision to find the energy efficient path to reach the destination nodes. Due to this process no loss will be occur, it's also improving stability of the networks. And then the following fig 5.1.3,4,5 explaining about the ARS implementation in the ns2 simulator.

## V .SIMULATION RESULTS AND DISCUSSION:

### 5.1 CONSTRUCTION OF MESH TOPOLOGY:

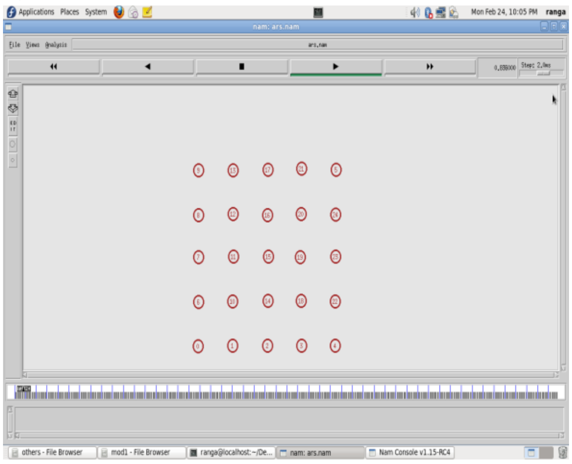

**Fig 5.1.1 shows topology construction in ns 2 shell**

### 5.2 SENDING HELLO PACKET TO THE NODES

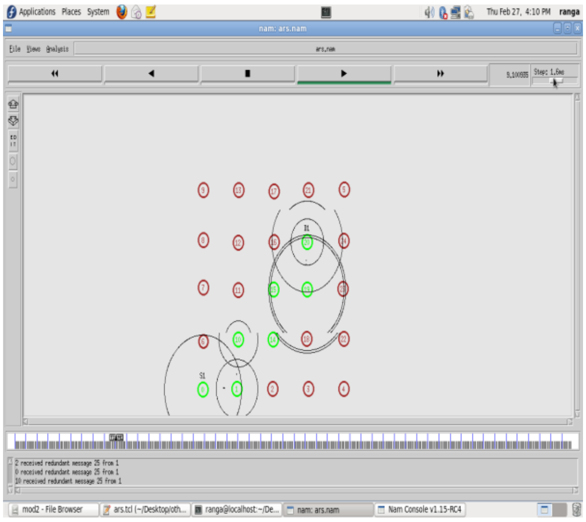

**Fig 5.2.1 shows hello packet transmission**

### 5.3 FIXING THE SOURCE AND DESTINATION NODE :

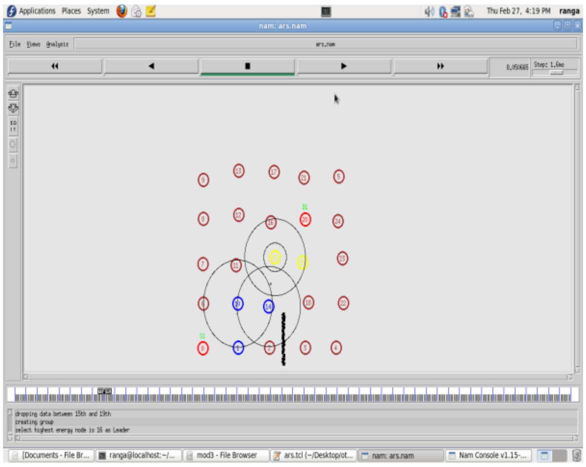

**Fig 5.3.1 shows node fixing for transmission of data's**

### 5.4 DATA LOSS DURING THE TRANSMISSION OF DATA'S

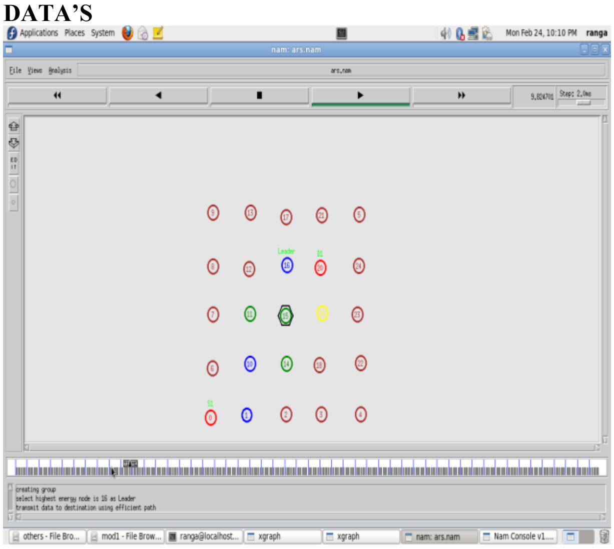

**Fig 5.4.1 shows data loss during the transmission due to the link and node failure**

### 5.5 HEADER LOOP CREATION :

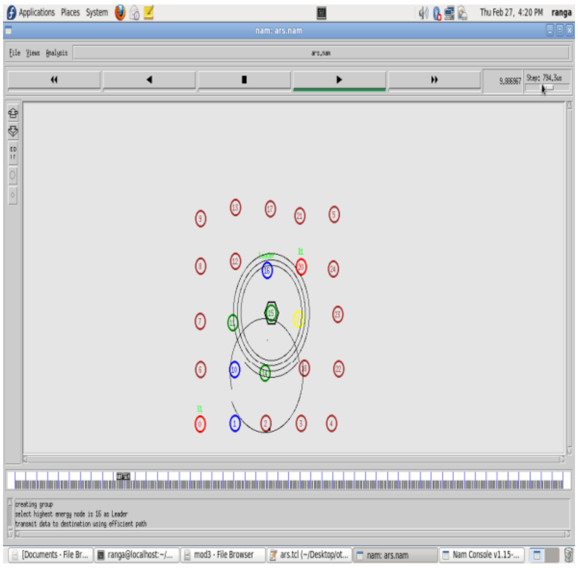

**Fig 5.4.1 shows creating the header loop to solve the problems**

## VI.GRAPH ANALYSIS:

### 6.1 THROUGHPUT ANALYSIS:

Throughput is the number of packets that is passing through the channel in a particular unit of time. This performance metric shows the total number of packets that have been successfully delivered from source node to the destination node and it can be improved with increasing node density. The following graph shows the ARS with DSDV protocol implementation of the throughput range. Based on the throughput, range the network performance will improve. Using the following formula we can find the value of throughput.

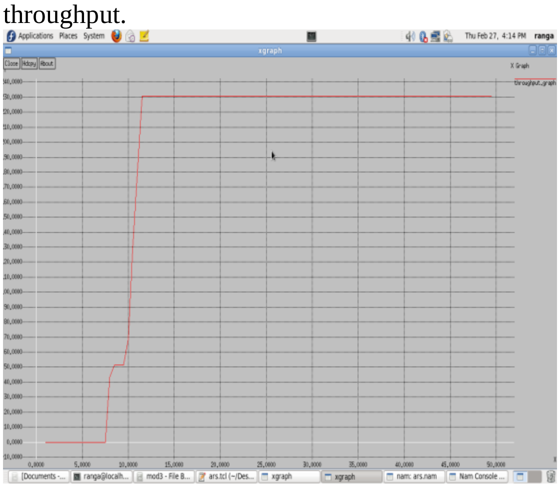

**Fig 6.1.1 throughput analysis of ARS with DSDV in ns2**

$$\text{Throughput} = \frac{\text{No. Of packets in bits}}{\text{Time}} \times 100$$

### 6.2 PACKET DROP ANALYSIS:

This is the other one factor of suggesting the network performance. Based on the packet drops on the networks the performance of the process will be efficient .In this concept of packet drops, reducing by the way maintaining or construction of the networks using ARS and DSDV protocols. The following graph showing the how much packet delivery on the destination nodes without any loss of data's. Once the failure occurs in the networks it will retransmit the data packet to the concern node. Because of these process losses is reduced .

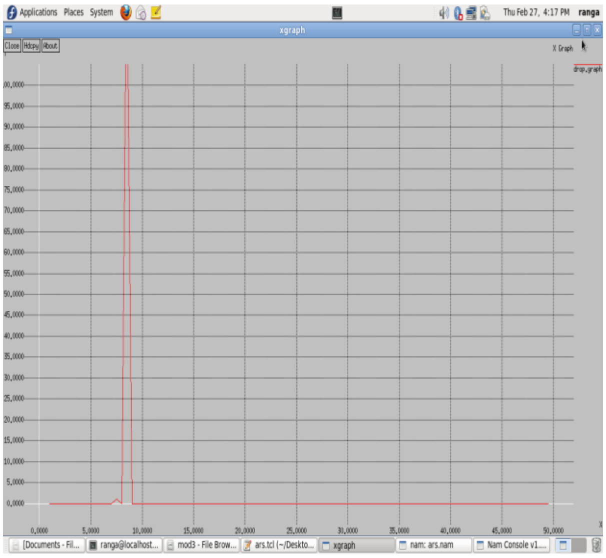

**Fig 6.2.1 shows the packet drop analysis of ARS with DSDV protocol**

## VII.CONCLUSION:

This paper presented an autonomous network reconfiguration system (ARS) with DSDV that enables a multi radio, wireless network to autonomously recover from wireless link failures. ARS generates an effective reconfiguration plan that requires only local network con-figuration changes by exploiting channel, radio, and path diversity. And finally the DSDV protocol contains the fluctuation problem in the wireless networks because of this can't obtain the cenpercent results.We can change the protocol implementation in the simulation the results will be efficient compare to this paper. This will be the future work and furthermore, ARS effectively identifies reconfiguration plans that satisfy applications' QoS constraints, admitting up to two times more flows than static assignment, through QoS-aware planning. Next, ARS's online reconfigurability allows for real-time failure detection and network reconfiguration, thus improving channel efficiency by 92%. Our experimental evaluation on a Linux-based implementation and ns2-based simulation has demonstrated the effectiveness of ARS in recovering from local link-failures and in satisfying applications' diverse QoS demands.

## REFERENCES;

[1] I. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: A survey," *Comput. Netw.*, vol. 47, no. 4, pp. 445–487, Mar. 2005.
[2] "MIT Roofnet," [Online]. Available: http://www.pdos.lcs.mit.edu/ roofnet
[3] Motorola, Inc., "Motorola, Inc., mesh broadband," Schaumburg, IL [Online]. Available: http://www.motorola.com/mesh
[4] P. Kyasanur and N. Vaidya, "Capacity of multi-channel wireless net-works: Impact of number of channels and interfaces," in *Proc. ACM MobiCom*, Cologne, Germany, Aug. 2005, pp. 43–57.
pp. K. Ramanchandran, E. Belding-Royer, and M. Buddhikot, "Inter-ference-aware channel assignment in multi-radio wireless mesh networks," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, 1–12.
[5] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. ACM MobiCom*, Philadelphia, PA, Sep. 2004, pp. 114–128.
[6] 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. ACM MobiCom*, Philadelphia, PA, Sep. 2004, pp. 216–230.
[7] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *Proc. ACM SIG-COMM*, Portland, OR, Aug. 2004, pp. 121–132.
[8] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, "Self-management in chaotic wireless deployments," in *Proc. ACM MobiCom*, Cologne, Germany, Sep. 2005, pp. 185–199.
[9] J. Zhao, H. Zheng, and G.-H. Yang, "Distributed coordination in dy-namic spectrum allocation networks," in *Proc. IEEE DySPAN*, Balti-more, MD, Nov. 2005, pp. 259–268.
[10] M. J. Marcus, "Real time spectrum markets and interruptible spectrum: New concepts of spectrum use enabled by cognitive radio," in *Proc. IEEE DySPAN*, Baltimore, MD, Nov. 2005, pp. 512–517.
[11] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *Proc. ACM MobiCom*, Cologne, Germany, Aug. 2005, P.P.58–72.
[12] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *Proc. ACM MobiCom*, Cologne, Germany, Aug. 2005, pp. 73–87.
[13] Brzezinski, G. Zussman, and E. Modiano, "Enabling distributed throughput maximization in wireless mesh networks: A partitioning approach,
[15] A. Raniwala and T. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proc. IEEE IN-FOCOM*, Miami, FL, Mar. 2005, vol. 3, pp. 2223–2234.
[16] S. Nelakuditi, S. Lee, Y. Yu, J. Wang, Z. Zhong, G. Lu, and Z. Zhang, "Blacklist-aided forwarding in static multihop wireless networks," in *Proc. IEEE SECON*, Santa Clara, CA, Sep. 2005, pp. 252–262.
[17] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1488–1505, Aug. 1999.
[18] L. Qiu, P. Bahl, A. Rao, and L. Zhou, "Troubleshooting multi-hop wireless networks," in *Proc. ACM SIGMETRICS*, Jun. 2005, pp. 380–381.
[19] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "experimental evaluation of wireless simulation assumptions," Dept. Comput. SCI., Dartmouth College, Hanover, NH, Tech. Rep. TR2004-507, 2004.
[19] Perkins, Charles E., Bhagwat Pravin: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, London England UK, SIGCOMM 94-8/94. < http://www.cise.ufl.edu/~helal/6930F01/papers/DS DV.pdf>
[20] Valid Nazari Talooki, and Jonathan Rodriguez, "Quality of Service for Flat Routing Protocols in Mobile Ad-hoc Network," ICST, 7-9 September 2009.
[21] Kiess W, Mauve M. 2007. A survey of real-world implementations of mobile ad-hoc networks. Ad Hoc Networks: 408-413.
[22] Marina MK, Das SR. 2001. On-demand multipath distance vector routing in ad hoc networks. Proceedings of the 9th IEEE International Conference on Network Protocols (ICNP).
[23] Perkins CE, Bhagwat P. 1994. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. Paper presented at the Proceedings of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications.
[23] Perkins CE. 1997. Ad hoc On demand Distance Vector (AODV) routing. IETF Internet draft, http://www.ietf.org/internet-drafts/draftietf-manet-aodv-00.txt.
[24] Ramanathan R, Redi J. 2002. A Brief Overview of Ad Hoc Networks: Challenges and Directions. IEEE Communications Magazine: 98-102.