



Inclusion of Flow Management for Automatic and Dynamic Route Discovery System by ARS

K.G.S. Venkatesan*

Associate Professor

Department of C.S.E

Bharath University, Chennai-600 073, INDIA

V. Khanaa

Dean of Information Technology

Dept. of Information Technology

Bharath University, Chennai, Tamil Nadu

Abstract— This paper presents an autonomous network reconfiguration system (ARS) that enables a multiradio WMN to autonomously recover from local link failures to preserve network performance. These failures cause severe performance deprivation in WMNs or require expensive manual network management for their real-time recovery. By using channel and radio diversities in WMNs, 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 considerably reconfigures network settings among local mesh routers. Multihop wireless Mesh Networks (WMN) experience frequent link failures due to channel interference, dynamic obstacles and bandwidth demands. They guide to performance deprivation in wireless mesh networks. WMNs are developing and deploy for public safety, environment monitoring and city wide wireless internet services.

Keywords— Multiradio Wireless Mesh Network mr-WMN, Autonomous network Reconfiguration System (ARS), Multihop wireless Mesh Network (WMN).

I. INTRODUCTION

The presentation of paper is under the domain of wireless mesh networks. Wireless mesh network having simple functions that enables the communication for the entire network. In this part we are discussing the paper domain and paper concept. Wireless mesh network created through the connection of wireless access points installed at each network user's setting. Each network user is also a provider, forwarding data to the next node. The networking infrastructure is decentralized and simplified because each node need only transmit as far as the next node. Wireless mesh networking could allow people living in remote areas and small businesses operating in rural neighborhoods to connect their networks together for affordable Internet connections.

In a full mesh topology, every node communicates with every other node, not just back and forth to a central router. In another variation, called a partial mesh network, nodes communicate with all nearby nodes, but not distant nodes. All communications are between the clients and the access point servers. The client/server relationship is the basis for this technology. A wireless mesh network can be seen as a special type of wireless ad-hoc network. A wireless mesh network often has a more planned configuration, and may be deployed to provide dynamic and cost effective connectivity over a certain geographic area [1]. An ad-hoc network, on the other hand, is formed ad hoc when wireless devices come within communication range of each other.

The mesh routers may be mobile, and be moved according to specific demands arising in the network. Often the mesh routers are not limited in terms of resources compared to other nodes in the network and thus can be exploited to perform more resource intensive functions. In this way, the wireless mesh network differs from an ad-hoc network, since these nodes are often constrained by resources.

A. Operation

The principle is similar to the way packets travel around the wired Internet— data will hop from one device to another until it reaches its destination. Dynamic routing algorithms implemented in each device allow this to happen. To implement such dynamic routing protocols, each device needs to communicate routing information to other devices in the network. Each device then determines what to do with the data it receives — either pass it on to the next device or keep it, depending on the protocol. The routing algorithm used should attempt to always ensure that the data takes the most appropriate (fastest) route to its destination.

B. Applications

Mesh networks may involve either fixed or mobile devices. The solutions are as diverse as communication needs, for example in difficult environments such as emergency situations, tunnels, oil rigs, battlefield surveillance, high speed mobile video applications on board public transport or real time racing car telemetry. An important possible application for wireless mesh networks is VoIP. By using a Quality of Service scheme, the wireless mesh may support local telephone calls to be routed through the mesh.

C. Some Current Applications

- U.S. military forces are now using wireless mesh networking to connect their computers, mainly ruggedized laptops, in field operations.
- Electric meters now being deployed on residences transfer their readings from one to another and eventually to the central office for billing without the need for human meter readers or the need to connect the meters with cables.
- The laptops in the one laptop per child program use wireless mesh networking to enable students to exchange files and get on the Internet even though they lack wired or cell phone or other physical connections in their area.
- The 66-satellite Iridium constellation operates as a mesh network, with wireless links between adjacent satellites. Calls between two satellite phones are routed through the mesh, from one satellite to another across the constellation, without having to go through an earth station [2]. This makes for a smaller travel distance for the signal, reducing latency, and also allows for the constellation to operate with far fewer earth stations that would be required for 66 traditional communications satellites.
- The Commotion Wireless Paper proposes building a 'device-as-infrastructure' distribution cryptic communications platform

D. Potential Advantages of Wireless Mesh Networks Includes

- Decreased need for Internet gateways
- Collaborative redundant backup technology, which insures data security in the event of disk failure
- The ability to configure routes dynamically
- Lower power requirements, which could potentially be met by low-cost or renewable energy sources
- Increased reliability: Each node is connected to several other nodes and if one drops out of the network, its neighbors simply find another route.

II. PAPER BACKGROUND

In this paper we are discussing self reconfiguration from failures is significance concept. One more thing we will attain higher throughput and high channel efficiency.

A. Need for Reconfiguration

Maintaining the performance of WMNs in the face of dynamic link failures remains a challenging problem. However, such failures can be withstood (hence maintaining the required performance) by enabling mr-WMNs to autonomously reconfigure channels and radio assignments. Recover from following failures

B. Recovering from Link-quality Degradation

The quality of wireless links in WMNs can degrade (i.e., *link-quality failure*) due to severe interference from other collocated wireless networks.

C. Satisfying dynamic QoS Demands

Links in some areas may not be able to accommodate increasing QoS demands from end-users (*QoS failures*), depending on spatial or temporal locality. By re-associating their radios/channels with underutilized radios/channels available nearby, links can avoid communication failures.

D. Coping with heterogeneous channel availability

Links in some areas may not be able to access wireless channels during a certain time period (*spectrum failures*) due to spectrum etiquette or regulation. Motivated by these three and other possible benefits of using reconfigurable mr-WMNs, in the remainder of this paper, we would like to develop a system that allows mr-WMNs to autonomously change channel and radio assignments (i.e., *self-reconfigurable*) to recover from the channel-related link failures mentioned.

III. 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 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 neighbouring mesh routers in addition to the faulty link(s).

Third, fault-tolerant routing protocols, such as local rerouting or multi path 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.

IV. PROPOSED SYSTEM

We propose an Autonomous Network reconfiguration System (ARS) that allows a multiradio 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 prepared with a reconfiguration planning algorithm that identifies local configuration changes for the recovery while minimizing changes of healthy network settings.

- ❖ 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 for the healthy network settings.
- ❖ Next, ARS also includes a monitoring protocol that enables a WMN to perform real-time failure recovery in conjunction with the planning algorithm [3]. 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”).
- ❖ Based on the measurement information, ARS detects link failures and/or generates QoS-aware network reconfiguration plans upon detection of a link failure. ARS has been implemented and evaluated extensively via experimentation on our multi radio WMN test-bed as well as via Java. Our evaluation results show that ARS outperforms existing failure-recovery methods, such as static or greedy channel assignments, and local rerouting. 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.

V. ARS ARCHITECTURE

ARS allows a multiradio WMN (mr-WMN) to autonomously reconfigure its local network settings channel, radio, and route assignment for real-time recovery from link failures [4]. 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 identifies reconfiguration plans that require the minimum number of changes for the healthy network settings. ARS also includes 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 neighbouring links (or “ripple effects”). In this paper, we using **Autonomous network Reconfiguration System algorithm** to achieve our goals. This algorithm has following steps to implement the target. ARS have the following steps

- Step1: Monitoring period (tm).
- Step2: Failure detection and group formation period (tf).
- Step3: Planning period (M, tf)).
- Step4: Reconfiguration period (P, tr).

Algorithm describes the operation of ARS. First, ARS in every mesh node monitors the quality of its outgoing wireless links at every (e.g., 10 s) and reports the results to a gateway via a management message. Second, once it detects a link Failure, ARS in the detector node(s) triggers the formation of a group among local mesh routers that use a faulty channel, and one of the group members is elected as a leader using the well-known bully algorithm for coordinating the reconfiguration. Third, the leader node sends a planning-request message to a gateway. Then, the gateway synchronizes the planning requests if there are multiple requests and generates a reconfiguration plan for the request. Fourth, the gateway sends a reconfiguration plan to the leader node and the group members. Finally, all nodes in the group execute the corresponding configuration changes, if any, and resolve the group. We assume that during the formation and reconfiguration, all messages are reliably delivered via a routing protocol and per-hop retransmission timer.

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.

A. Overview

ARS is a distributed system that is easily deployable in IEEE 802.11-based mr-WMNs. running in every mesh node, ARS supports self-reconfigurability via the following distinct features.

B. Localized reconfiguration

Based on multiple channels and radio associations available, ARS generates reconfiguration 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.

C. *QoS-aware planning*

ARS effectively identifies QoS-satisfiable reconfiguration plans by:

- 1) Estimating the QoS-Satisfiability of generated reconfiguration plans; and
- 2) Deriving their expected benefits in channel utilization.

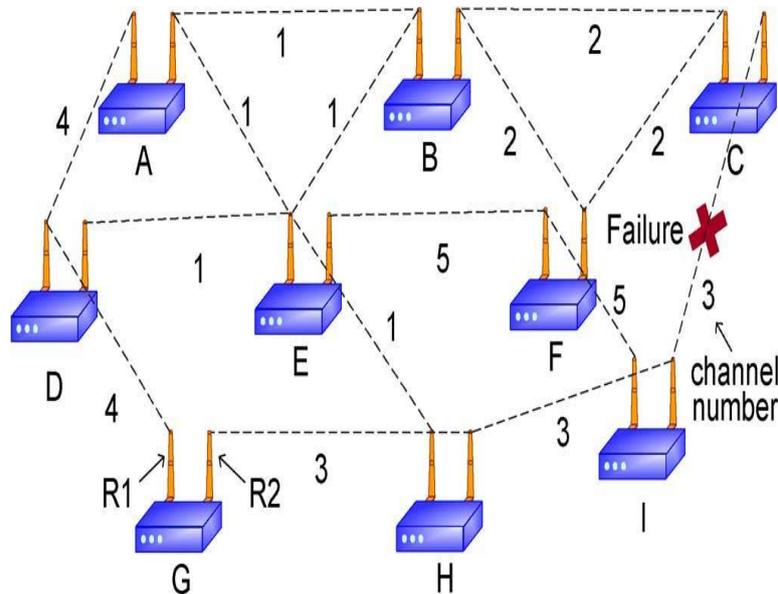


Fig. 1 Multi_hop Wireless Mesh network channel numbers & failures.

D. *Autonomous Reconfiguration via Link-quality Monitoring*

ARS accurately monitors the quality 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.

E. *Cross-layer Interaction*

ARS actively interacts 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 recovery period with the help of a routing protocol.

Algorithm 1 describes the operation of ARS. First, ARS in every mesh node monitors the quality of its outgoing wireless links at every (e.g., 10 s) and reports the results to a gateway via a management message. Second, once it detects a link failure(s), ARS in the detector node(s) triggers the formation of a group among local mesh routers that use a faulty channel, and one of the group members is elected as a leader using the well-known bully algorithm [9] for coordinating the reconfiguration. Third, the leader node sends a planning-request message to a gateway. Then, the gateway synchronizes the planning requests—if there are multiple requests—and generates a reconfiguration plan for the request. Fourth, the gateway sends a reconfiguration plan to the leader node and the group members. Finally, all nodes in the group execute the corresponding configuration changes, if any, and resolve the group. We assume that during the formation and reconfiguration, all messages are reliably delivered via a routing protocol and per-hop retransmission timer. In what follows, we will detail each of these operations, including how to generate reconfiguration plans, how to monitor link conditions such as bandwidth, and how much overhead ARS generates for the monitoring and for maintaining a reconfiguration group.

F. *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 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 reconfiguration 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, 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.

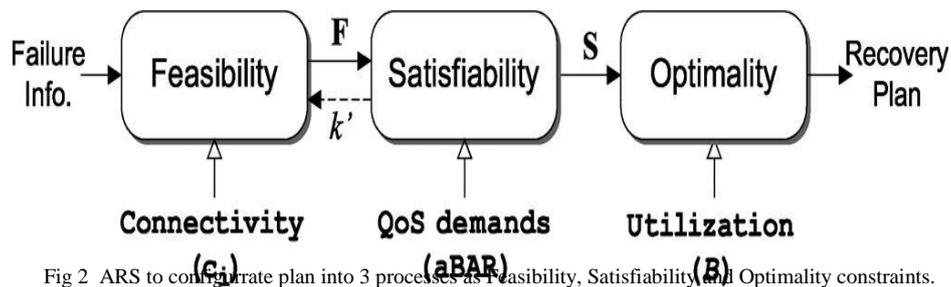


Fig 2 ARS to configure plan into 3 processes as Feasibility, Satisfiability, and Optimality constraints.

Algorithm 1: ARS Operation at mesh node i of Monitoring period, Failure detection, Planning period & Reconfiguration periods.

- (1) Monitoring period (t_m)
 - 1: **for every** link j **do**
 - 2: measure link-quality (l_q) using passive monitoring;
 - 3: **end for**
 - 4: send monitoring results to a gateway g ;
- (2) Failure detection and group formation period (t_f)
 - 5: **if** link l violates link requirements r **then**
 - 6: request a group formation on channel c of link l ;
 - 7: **end if**
 - 8: participate in a leader election if a request is received;
- (3) Planning period (M, t_p)
 - 9: **if** node i is elected as a leader **then**
 - 10: send a planning request message (c, M) to a gateway;
 - 11: **else if** node i is a gateway **then**
 - 12: synchronize requests from reconfiguration groups M_n
 - 13: generate a reconfiguration plan (p) for M_i ;
 - 14: send a reconfiguration plan to a leader of M_i ;
 - 15: **end if**
- (4) Reconfiguration period (p, t_r)
 - 16: **if** p includes changes of node i **then**
 - 17: apply the changes to links at t ;
 - 18: **end if**
 - 19: relay p to neighboring members, if any

G. Feasible Plan Generation

Generating feasible plans is essentially to search all legitimate changes in links' configurations and their combinations around the faulty area. Given multiple radios, channels, and routes, ARS identifies feasible changes that help avoid a local link failure but maintain existing network connectivity as much as possible. However, in generating such plans, ARS has to address the following challenges.

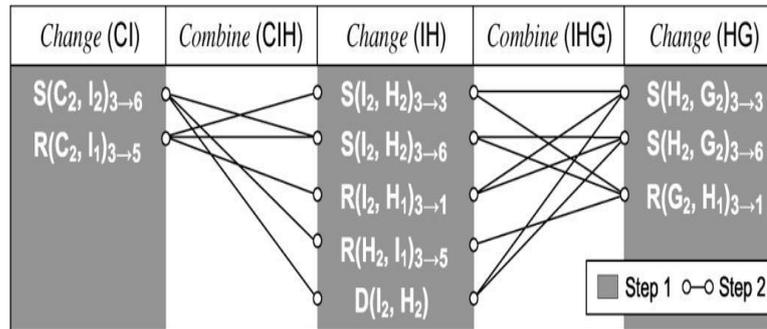
H. Avoiding a faulty channel

ARS first has to ensure that the faulty link needs to be fixed via reconfiguration. To this end, ARS considers three primitive link changes, as explained in Table I. Specifically, to fix a faulty link(s), ARS can use:

- 1) A channel-switch where both end-radios of link AB can simultaneously change their tuned channel;
- 2) A radio-switch where one radio in node A can switch its channel and associate with another radio in node B; and
- 3) A route-switch where all traffic over the faulty link can use a detour path instead of the faulty link.

TABLE I

LINK-CHANGE IN ARS. A CHANGE WILL REPRESENTS A PRIMITIVE CHANGE AND THEIR DESCRIPTION WILL BE EXPLAINED CLEARLY.



Examples of feasible plans generated:

$$P_1=[S(C_2, l_2)_{3-6}, S(l_2, H_2)_{3-6}, S(H_2, G_2)_{3-6}], P_2=[S(C_2, l_2)_{3-6}, D(l_2, H_2), S(H_2, G_2)_{3-6}], \dots, P_{11}$$

I. Maintaining Network Connectivity and Utilization

While avoiding the use of the faulty channel, ARS needs to maintain connectivity with the full utilization of radio resources. Because each radio can associate itself with multiple neighboring nodes, a change in one link triggers other neighbouring links to change their settings. To coordinate such propagation, ARS takes a two-step approach. ARS first generates feasible changes of each link using the primitives, and then combines a set of feasible changes that enable a network to maintain its own connectivity. Furthermore, for the combination, ARS maximizes the usage of network resources by making each radio of a mesh node associate itself with at least one link and by avoiding the use of same (redundant) channel among Radios in one node.

J. Controlling the Scope of Reconfiguration changes

ARS has to limit network changes as local as possible, but at the same time it needs to find a locally optimal solution by considering more network changes or scope. To make this tradeoff, ARS uses a -hop reconfiguration parameter. Starting from a faulty link(s), ARS considers link changes within the first hops [6] and generates feasible plans. If ARS cannot find a local solution, it increases the number of hops so that ARS may explore a broad range of link changes. Thus, the total number of reconfiguration changes is determined on the basis of existing configurations around the faulty area as well as the value. Given the failure in link CI, ARS first generates feasible and desirable changes per link (gray columns) using the primitives. Here, the changes must not include the use of a faulty or redundant channel. Next, ARS combines the generated per-link primitives of neighboring links to generate a set of feasible plans.

Primitive changes	Description
Channel switch ($S(A_i, B_j)_{\alpha \rightarrow \beta}$)	Radios A_i and B_j of link AB switch their channel (α) to other channel (β).
Radio switch ($R(A_i, B_j)_{\alpha \rightarrow \beta}$)	Radio A_i in node A re-associates with radio B_j in node B , tuned in channel (β).
Detouring ($D(A_i, B_j)$)	Both radios A_i and B_j of link AB remove their associations and use a detour path, if exists.

Fig 3 ARS to generate examples of feasible plan generated by change (CI) and combine (CIH).

K. Per-link Bandwidth Estimation

For each feasible plan, ARS has to check whether each link's configuration change satisfies its bandwidth requirement, so it must estimate link bandwidth. To estimate link bandwidth, ARS accurately measures each link's capacity and its available channel airtime. In multihop wireless networks equipped with a CSMA-like MAC, each link's achievable bandwidth (or throughput) can be affected by both link capacity and activities of other links that share the channel airtime. Even though numerous bandwidth-estimation techniques have been proposed, they focus on the average bandwidth of each node in a network or the end-to-end throughput of flows which cannot be used to calculate the impact of per-link configuration changes. By contrast, ARS estimates an individual link's capacity based on measured (or cached) link-quality information—packet-delivery ratio and data-transmission rate measured by passively monitoring the

transmissions of data or probing packets [10]—and the formula derived in the Appendix. Here, we assume that ARS is assumed to cache link-quality information for other channels and use the cached information to generate reconfiguration plans. If the information becomes obsolete, ARS detects link failures and triggers another reconfiguration to find QoS-satisfiable plans—lazy monitoring.

L. Examining per-link Bandwidth Satisfiability

Given measured bandwidth and bandwidth requirements, ARS has to check if the new link change(s) satisfies QoS requirements. ARS defines and uses the expected busy airtime ratio of each link to check the link's QoS satisfiability.

VI. MODULES

In this paper we construct three modules for attaining our goals. They are follows as

A. Identifies Reconfiguration Plan

We evaluated the impact of the reconfiguration range. We used the same experiment settings as the previous one and focused on reconfiguration requests at *TI*. As we increase the hop count from a faulty link(s), we measure the capacity improvement achieved by the reconfiguration plans. In addition, we calculate the capacity gain per change as the cost-effectiveness of reconfiguration planning with different *k* values.

B. Avoid Ripple Effect

We also studied ARS's effectiveness in avoiding the ripple effects of network reconfiguration. In this topology, we run six UDP flows(f_1, \dots, f_6) each at 4 Mb/s and measure each flow's throughput while injecting interference into a target channel. We run the same scenarios with two different interference frequencies (5.28 and 5.2 GHz) to induce failures on different links [8]. Also, we use three failure-recovery methods (i.e., local rerouting, greedy, and ARS) for comparison.

C. Local Reconfiguration

ARS's avoidance of ripple effects. ARS finds a local reconfiguration plan that avoids the ripple effects by considering neighbouring nodes' channel utilization, whereas the greedy channel switching and local rerouting cannot fully recover from the failure or cause additional QoS failures. On the detection of the interference, ARS switches the channel of all the fault-related links, based on its planning algorithm to another channel.

Naturally, this result (configuration and throughput) is the same as that achieved by the greedy method. On the other hand, the local rerouting causes heavy channel contention for detour paths, degrading neighbouring flows' performance as well as others.

VII. EXPERIMENTAL SETUP

To evaluate our implementation, we constructed a multi-hop wireless mesh network testbed. The testbed consisting of 18 mesh nodes and have multiple 5 links. Each node is deliberately placed on either ceiling panels or high-level shelves to send/receive strong signals to neighbouring nodes. On the other hand, each node will experience enough multi-path fading effects from obstacles and interferences from co-existing public wireless networks.

VIII. EXPERIMENTAL RESULTS

In this paper, we evaluated the improvements achieved by ARS, including throughput and channel efficiency, QoS satisfiability and reduction of ripple effects. We study throughput and channel-efficiency gains via ARS real-time reconfiguration. We run one UDP flow at a maximum rate over a randomly-chosen link in our testbed, while increasing the level of interference every 10 seconds. We also set the QoS requirements of every link to 6 Mbps, and measure the flows throughput progression every 10 seconds during a 400 seconds run. For the purpose of comparison, we also ran the same scenario under the local re-routing with a WCETT (Weighted Cumulative Expected Transmission Time) metric and a static channel assignment algorithms.

The progression of link throughput achieved by the above methods, ARS effectively reconfigures the network on detection of a failure, achieving 457% and 277% more bandwidth than static-assignment and local re-routing, respectively. ARS accurately detects a links QoS-failure using link-quality monitoring information, and completes network reconfiguration within 16 seconds on average, while the static-assignment experiences severe throughput degradation.

ARS also improves channel efficiency by more than 92 % over the other recovery methods. Using the data collected during the previous experiments, we derive channel efficiency of the UDP flow by counting the number of successful transmissions.

IX. PERFORMANCE EVALUATION

In this paper, we evaluated ARS in large-scale network settings via simulation [8]. In the simulation, we use a grid topology with 30 nodes. In the topology, adjacent nodes are separated by 180m and each node is equipped with a different number of radios [9], depending on its proximity to a gateway. The gateway is equipped with a four radios, one-hop-away nodes from a gateway have three radios, and other nodes have two radios.

For each node in the above topology, we use the following network protocol stack. First, the shadowing propagation model [12] is used to simulate varying channel quality and multi-path effects. Next, CMU 802.11 wireless extension is used for the MAC protocol with a fixed data rate and further modified to support multiple radios and multiple channels.

X. EVALUATION RESULTS

We measured the effectiveness of ARS in meeting the varying QoS requirements in a mr-WMN. We initially assign symmetric link capacity in the channel assignment of the grid topology. ARS reconfigures a wireless mesh network to meet different QoS requirements. Before each configuration, the gray areas can only accept 1 to 10 UDP flows. On the other hand, after reconfiguration, the network in the average network in the areas can admit 4 to 16 additional flows, improving the average network capacity of the gray areas by 4 times.

We evaluated the impact of the configuration range. We used the same experiment setting as the previous one and focused on reconfiguration requests at T1. As we increase the hop count k from a faulty links, we measure the capacity improvement achieved by the configuration plans. In additional, we calculate the capacity gain per change as the cost-effectiveness of reconfiguration planning with different k values.

XI. FUTURE WORK

ARS is mainly evaluated in IEEE 802.11a networks, where 13 orthogonal channels are available. However, ARS can also be effective in a network with a small number of orthogonal channels [1]. Because ARS includes a link association primitives, it can learn available channel capacity by associating with idle interfaces of neighbouring nodes, and it further limits the range of a reconfiguration group.

XII. CONCLUSIONS

This paper presents an Autonomous Network reconfiguration System (ARS) that enables a multi-radio WMN to autonomously recover from wireless link failures. ARS generates an effective reconfiguration plan that requires only local network configuration changes by exploiting channel, radio, and path diversity.

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.

ACKNOWLEDGMENT

The author would like to thank the Director, Secretary, Correspondent, Principal, HOD, Dean CSE of Bharath University, Chennai for their motivation and constant encouragement. The author would like to specially thank Dr. V.Khanna for his guidance and for critical review of this manuscript and for his valuable input and fruitful discussions in completing the work and the Faculty Members of Department of Computer Science & Engineering. Also, he takes privilege in extending gratitude to his parents and family members who rendered their support throughout this Research work.

REFERENCES

- [1] K. Ramachandran, E. Belding-Royer, and M. Buddhikot, "Interference-Aware channel assignment in multi-radio wireless mesh Networks," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–12.
- [2] Jing Shi, Rui Zhang, Yanchao Zhang IEEE January 2011.
- [3] 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.
- [4] 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.
- [5] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level Measurements from an 802.11b mesh network," in *Proc. ACM SIGCOMM*, Portland, OR, Aug. 2004, pp. 121–132.
- [6] 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.
- [7] J. Zhao, H. Zheng, and G.-H. Yang, "Distributed coordination in dynamic spectrum allocation networks," in *Proc. IEEE DySPAN*, Baltimore, MD, Nov. 2005, pp. 259–268.
- [8] "The network simulator—ns-2," [Online]. Available: <http://www.isi.edu/nsnam/ns>.
- [9] T. S. Rappaport, "Wireless Communications: Principles and Practice", Englewood Cliffs, NJ: Prentice-Hall, 2002.
- [10] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced Distance-vector routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM*, London, U.K., Sep. 1994, pp. 234–244.
- [11] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.
- [12] T.S. Rappaport, "Wireless Communications : Principles and Practice," Prentice Hall, 2002.
- [13] Netfilter, <http://www.netfilter.org>
- [14] MADWiFi, <http://www.madwifi.org>

- [15] Atheros Communications, <http://www.atheros.com>
- [16] Soekris Engineering, <http://www.soekris.com>
- [17] A.S. Tanenbaum and M.V. Steen, "Distributed Systems," Pearson Education.

ABOUT THE AUTHOR



K.G.S.Venkatesan received his B.Tech degree in Computer Science & Engineering from JNT University, Hyderabad and received his M.Tech degree in Computer Science & Engineering from Bharath University. He is currently pursuing his Ph.D in Computer Science & Engineering at Bharath University, Chennai. He has 10 years of Teaching experience and has guided many B.Tech and M.Tech projects. He is having Membership in Indian Society of Technical Education (MISTE).He attended Teaching Skills Programme conducted by WIPRO MXLA (Mission 10X Learning Approach).



Dr.V.KHANAA received his B.E and M.S degree in Computer Science & Engineering from BITS Pilani and has obtained the **Ph.D** degree from Bharath University. He is currently working as Dean-Information in Bharath University, Chennai, Tamilnadu, India and his research interest are Computer Networks, Digital Image Processing. He is having Membership in Indian Society of Technical Education (MISTE).