Detection of Link Failures and Autonomous Reconfiguration in WMNs

Sumathi P[1] Srinivasulu M[2]
1.Dept of CSE(M.Tech,Madanapalle Institute of Technology and Science, Madanapalle
2.Dept of CS & SE(M.Tech, AUCE (A), Visakhapatnam, AP, India
Suma.cse51@gmail.com, m.seenu@yahoo.co.in

Abstract

During their lifetime, multihop wireless mesh networks (WMNs) experience frequent link failures caused by channel interference, dynamic obstacles, and/or applications' bandwidth demands. By reconfiguring these link failures ARS generates an effective reconfiguration plan that requires only local network configuration changes by exploiting channel, radio, and path diversity. ARS effectively identifies reconfiguration plans that satisfy QoS constraints. And ARS's online reconfigurability allows for real-time time failure detection and network reconfiguration. ARS is mainly evaluated in IEEE 802.11a networks. It's design goal is to reconfigure from network link failures accurately. Even then WMNs face some frequent link failures. By overcome these problems we present Localized sElf-reconfiGuration algOrithms (LEGO) to autonomously and effectively recnfigure from wireless link failures. First, LEGO locally detects link failures. Second, it dynamically forms/deforms a local group for cooperative network reconfiguration among local mesh routers in a fully distributed manner. Next, LEGO intelligently generates a local network reconfiguration plan. Finally, by figuring local channel utilization and reconfiguration cost in its planning, LEGO maximizes the network's ability to meet diverse links' QoS demands. LEGO has been implemented on a Linux-based system and experimented on a real life test bed, demonstrating its effectiveness in recovering from link failures and its improvement of channel efficiency by up to 92%.

Keywords - Self-Reconfigurable Networks, Multi-Radio Wireless Networks, IEEE 802.11, WLAN access points (AP).

I. INTRODUCTION

In WMNs, nodes are comprised of mesh routers and mesh clients. Each node operates not only as a host but also as a router, forwarding packets on behalf of other nodes that may not be within direct wireless transmission range of their destinations. A WMN is dynamically self-organized and self-configured, with the nodes in the network automatically establishing and maintaining mesh connectivity among themselves (creating, in effect, an ad hoc network). Routing in ad-hoc wireless networks has been an active area of research for many years. Much of the original work in the area was motivated by mobile application environments, such as battlefield ad hoc networks. The use of multiple radios is complementary to the use of directional antennas, and we believe that our protocol can be modified for directionality. Specifically, we would have to revisit the assumption that all same-channel links along a path interfere with one another. Another way to improve the capacity of a wireless network is to take advantage of the full spectrum by using rapid channel switching. This approach has been explored by several researchers. However, channel switching can be quite slow with existing 802.11 hardware. With the availability of better hardware, many of the proposed approaches based on rapid channel switching will become feasible. Our approach, however, works with currently available hardware. We also note that even with the ability to switch channels rapidly, a single radio can not transmit and receive simultaneously. Thus, the use of multiple radios can provide a performance improvement even in this case. A novel approach that takes advantage of the inherent multi-radio capability of WMNs. We show that this capability can enable partitioning of the network into sub networks in which simple distributed scheduling algorithms can achieve 100% throughput. The partitioning is based on the recently introduced notion of Local Pooling. Using this notion, we characterize topologies in which 100% throughput can be achieved distributedly. These topologies are used in order to develop a number of channel assignment algorithms that are based on a matroid intersection algorithm[5]. These algorithms partition a network in a manner that not only expands the capacity regions of the sub networks but also allows distributed algorithms to achieve these capacity regions. Finally, we evaluate the performance of the algorithms via simulation and show that they significantly increase the distributedly achievable capacity region. Joint scheduling and routing in a slotted multihop wireless network with a stochastic packet arrival process was considered in the seminal paper by Tassiulas and Ephremides [11].

In that they presented the first centralized policy that is guaranteed to stabilize the network (i.e. provide 100% throughput) whenever the arrival rates are within the stability region. The results of [11] have been extended to various settings of wireless networks and input-queued switches. However, optimal algorithms based on [11] require repeatedly solving a global optimization problem, taking into account the queue backlog information for every link in the network. Obtaining a centralized solution to such a problem in a wireless network does not seem to be feasible, due to the communication overhead associated with continuously collecting the queue backlog information. On the other hand, distributed algorithms usually provide only approximate solutions, resulting in significantly reduced throughput. Using AODV and DSR routing Protocols the packets are reached to destination. Using DSR, the network is completely self-organizing and self-configuring, requiring no existing network infrastructure or administration. Network nodes (computers) cooperate to forward packets for each other to allow communication overmultiple “hops” between nodes not directly within wireless transmission range of one another. As nodes in the network move about or join or leave the network, and as wireless transmission conditions such as sources of interference change, all routing is automatically determined and maintained by the DSR routing protocol. Since the number or sequence of intermediate hops needed to reach any destination may change at any time, the resulting network topology may be quite rich and rapidly changing. The DSR protocol allows nodes to dynamically discover a *source route* across multiple network hops to any destination in the ad hoc network. By including this source route in the header of each data packet, other nodes forwarding or overhearing any of these packets may also easily cache this routing information for future use[8]. The original motivation in the design of DSR came from the operation of theAddressResolution Protocol (ARP) used in the TCP/IP suite of protocols in the Internet. ARP is used on Ethernets and other types of networks to find the link-layer MAC address of a node on the same subnet as the sender. A node sending a packet to a local IP address for which it does not yet have the MAC address cached, broadcasts an ARP REQUEST packet on the local subnet link, giving the IP address of the node it is looking for; that node responds with an ARP REPLY packet, giving its MAC address, and all other nodes ignore the REQUEST. If all nodes in an ad hoc network are within wireless transmission range of each other, this is the only routing protocol needed for the ad hoc network. ABR also adds overhead for periodic beacon packets required to monitor link stability. The Ad Hoc On-Demand Distance Vector routing protocol (AODV) uses mechanisms similar to DSR’s Route Discovery and Route Maintenance, but it uses them to create hop-by-hop routes rather than source routes as is done in DSR[8]; this use of hop-by-hop routes avoids the source routing header overhead of DSR but prevents or makes difficult many of the route caching and other Route Discovery optimizations present in DSR and prevents AODV from supporting uni-directional links between nodes. we establish the capacity of general multi channel networks wherein the number of interfaces, m, may be smaller than the number of channels, c.However, one important exception is a random network with up to O (log n) channels, independent of the number of interfaces available at each node[2]. This implies that it may be possible to build capacity-optimal multi-channel networks with as few as one interface per node. We also extend our model to consider the impact of interface switching delay, and show that capacity losses due to switching delay can be avoided by using multiple interfaces. ARS has been implemented and evaluated extensively via experimentation on our multiradio WMN 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 assignments, and local rerouting. ARS’s local reconfiguration improves network throughput and channel efficiency by more than 26% and 92%, respectively, over the local rerouting scheme.

II.PROBLEM DEFINITION

Wireless mesh networks have the potential to deliver Internet broadband access, wireless local area network coverage and network connectivity for stationary or mobile hosts at low costs both for network operators and customers.

We first describe the need for self-reconfigurable mr WMNs. Next, we introduce the network model and assumptions.

A. Why Is Self-Reconfigurability Necessary?

By enabling mr-WMNs to autonomously reconfigure channels and radio1 assignments, as in the following examples.

• *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.

• *Satisfying dynamic QoS demands*:

Links in some areas may not be able to accommodate increasing QoS demands from end-users depending on spatial or temporal locality.

• 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.

B. Network Model and Assumptions

- Multiradio WMN:

A network is assumed to consist of mesh nodes, IEEE 802.11-based wireless links, and one control gateway. Each mesh node is equipped with radios, and each radio’s channel and link assignments are initially made by using (e.g., see Fig. 1) global channel/link assignment algorithms Multiple orthogonal channels are assumed available.

- QoS Support:

During its operation, each mesh node periodically sends its local channel usage and the quality information for all outgoing links via management messages to the control gateway.

- Link Failures:

Channel-related link failures that we focus on are due mainly to narrowband channel failures. These failures are assumed to occur and last in the order of a few minutes to hours, and reconfiguration is triggered in the same order of failure occurrences.

III. SYSTEM ARCHITECTURE

Wireless mesh architecture is a first step towards providing cost effective and dynamic high-bandwidth networks over a specific coverage area. Wireless mesh architectures infrastructure is, in effect, a router network minus the cabling between nodes. It's built of peer radio devices that don't have to be cabled to a wired port like traditional WLAN access points (AP) do. Mesh architecture sustains signal strength by breaking long distances into a series of shorter hops. Intermediate nodes not only boost the signal, but cooperatively make forwarding decisions based on their knowledge of the network, i.e. perform routing. Such an architecture may with careful design provide high bandwidth, spectral efficiency, and economic advantage over the coverage area.

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 communication network 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.

- 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.
- QoS-aware planning: ARS effectively identifies QoS-satisfiable reconfiguration plans by: 1) estimating the QoSsatisfiability of generated reconfiguration plans; and 2) deriving their expected benefits in channel utilization.
- Autonomous reconfiguration via link-quality monitoring: ARS accurately monitors the quality4 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.
- 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.

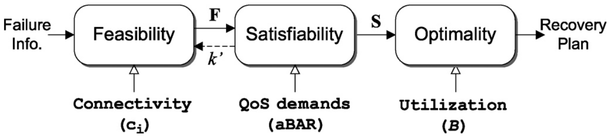


Fig. 1. Localized reconfiguration planning in ARS. ARS generates a reconfiguration plan by breaking down the planning process into three processes with different constraints.

B. Planning for Localized Network Reconfiguration

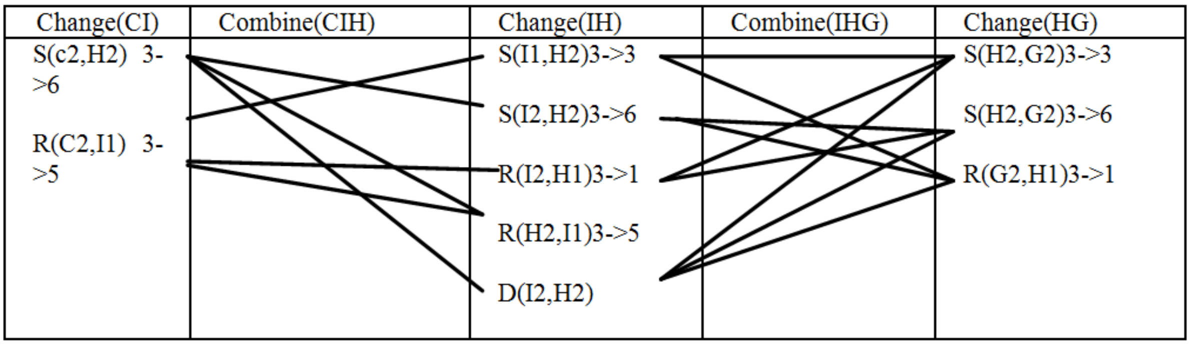
The core function of ARS is to *systematically* generate localized reconfiguration plans. A *reconfiguration plan* is defined as a set of links' configuration changes (e.g., channel switch, link association) necessary for a network to recover from a link(s) failure on a channel, and there are usually multiple reconfiguration plans for each link failure. 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. 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.

Feasible Plan Generation*:* Generating feasible plans is essentially to search all legitimate changes in links' configurations and their combinations around the faulty area. However, in generating such plans, ARS has to address the following challenges.

- Avoiding a faulty channel: ARS first has to ensure that the faulty link needs to be fixed via reconfiguration.

• 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 neighboring links to change their settings.

• 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 and generates feasible plans.



Examples of feasible plans generated

P1=[S(C2,I2)3->6,S(I2,H2)3->6,S(H2,G2)3->6],
P2=[S(C2,I2)3->6,D(I2,H2)S(H2,G2)3->3]…P11
Fig.2. Example of network planning.

Let us consider an illustrative example in Fig. 4. 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. During the combination, ARS has to preserve link and/or radio connectivities. After the two steps, ARS has 11 feasible reconfiguration plans(F) by traversing connected changes of all links considered in the planning.

QoS-Satisfiability Evaluation: Among a set of feasible plans,ARS now needs to identify QoS-satisfying reconfiguration plans by checking if the QoS constraints are met under each plan.

To filter out such plans, ARS has to solve the following challenges.

• *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

• *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.

Choosing the Best Plan: ARS now has a set of reconfiguration plans that are QoS-satisfiable and needs to choose a plan within the set for a local network to have evenly distributed link capacity.

C. Complexity of ARS

ARS incurs reasonable bandwidth and computation overheads. First, the network monitoring part in the reconfiguration protocols is made highly efficient by exploiting existing data traffic and consumes less than 12 kb/s probing bandwidth (i.e., one packet per second) for each radio. In addition, the group formation requires onlyO(n) message overhead (in forming a spanning tree), where n is the number of nodes in the in the group. Next, the computational overhead in ARS mainly stems from the planning algorithms. Specifically, generating its possible link plans incurs O(n+m) complexity, where n is the number of available channels and m is the number of radios. Next, a gateway node needs to generate and evaluate feasible plans, which incurs search overhead in a constraint graph that consists of O(l(n+m))nodes, where is the number of links that use a faulty channel in the group.

V CONCLUSION

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. Based on existing MAC, routing, and transport protocols, network performance is not scalable with either the number of nodes or the number of hops in the network. This problem can be alleviated by increasing the network capacity through using multiple channels/radios per node or developing wireless radios with higher transmission speed. However, these approaches do not truly enhance the scalability of WMNs, because resource utilization is not actually improved. Therefore, in order to achieve scalability, it is essential to develop new MAC, routing, and transport protocols for WMNs.

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. P. Kyasanur and N. Vaidya, "Capacity of multi-channel wireless networks: Impact of number of channels and interfaces," in *Proc. ACM MobiCom*, Cologne, Germany, Aug. 2005, pp. 43–57.
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. 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.
5. A. Brzezinski, G. Zussman, and E. Modiano, "Enabling distributed throughput maximization in wireless mesh networks: A partitioning approach," in *Proc. ACM MobiCom*, Los Angeles, CA, Sep. 2006, pp. 26–37.
6. A. Raniwala and T. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proc.IEEE INFOCOM*, Miami, FL, Mar. 2005, vol. 3, pp. 2223–2234.
7. 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.
8. D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *The Book of Mobile Computing*. Norwell, MA: Kluwer, 1996, vol. 353.
9. A. P. Subramanian, H. Gupta, S. R. Das, and J. Cao, "Minimum interference channel assignment in multiradio wireless mesh networks,"
10. K.-H. Kim and K. G. Shin, "On accurate and asymmetry-aware management of link quality in wireless mesh networks," *IEEE/ACMTrans. Netw.*, vol. 17, no. 4, pp. 1172–1185, Aug. 2009.

Authors Profile

Srinivasulu M received his M.Tech degree from Andhra University, India in 2012. And also i received M.C.A degree from S V University, India in 2008. He is having overall experience in System Engineering 1 year 8 months. He has completed successfully Research paper "**Finding the colour blindness using Ishihara algorithm**" and also "**completed Research paper Data clustering with leaders & sub leaders Algorithm**".

Sumathi P received her M.Tech degree from JNTU Anantapur,AP, India in 2012 and also I received B.Tech degree from JNTU Anantapur,AP,India in 2008. I attended IOCAS International Conference in May- 2012,