Throughput Enhancement through Efficient Channel Assignment in Multi-radio Ad Hoc Networks

Ashish Sangwan* **Ghanshyam Dass*** **Pradipta De**** **Huzur Saran***

sangwan@cse.iitd.ernet.in *cs1030163@cse.iitd.ernet.in* *pradipta.de@in.ibm.com* *saran@cse.iitd.ernet.in*

*Department of Computer Science, Indian Institute of Technology, Delhi 110016, India

**IBM, India Research Lab., Delhi 110016, India

Abstract - Use of a single wireless network interface on a device had been the standard practice for a long time due to hardware limitations. Designing a multi-hop ad hoc network with single NIC devices suffers from low bandwidth utilization due to co-channel interference. Recent advances in wireless device technology is spurring the emergence of several end user devices with multiple radios on them. Wireless access networks designed using these multi-radio devices has been shown to give manifold improvement in throughput compared to single channel networks. However, these works have mainly focused on a gateway-based mesh architecture. With growing number of multi-radio end user devices, we envision a future where pure ad hoc communication among these devices will become commonplace without a specific structure. This paper provides a distributed solution to the channel assignment problem to multiple NICs in such an ad hoc scenario. The well-known routing protocols used in single channel scenario, namely AODV is used for route selection, but the potential for an improved routing algorithm for multi-channel case has been highlighted. We also provide an interface switching strategy so that the flows are not starved. We evaluate our channel assignment scheme through simulations on GloMoSim.

Keywords: Distributed Algorithms/Protocols, Design, Routing, AODV, Wireless Ad-Hoc Networks

Categories

[3*c*] - **New signaling and control traffic in multimedia/multisystems** [4*f*] - **End to End service performance evaluation**

I. Introduction

Use of a single radio on a wireless access device had been a prevalent practice due to limitations of hardware and vendor support. Single radio networks suffer from low network throughput in multi-hop topology due to co-channel interference. This has spurred the recent research into using multiple radios on a single wireless device. Each network interface card (NIC) is assigned a non-overlapping channel enabling simultaneous interference-free transmission. Use of multiple NICs along with appropriate channel assignment can increase the overall network throughput manifold [4, 7, 9], and have been applied in architecting high-speed enterprise backbones based on multi-hop multi-NIC mesh topology.

Majority of the work in multi-channel multi-radio networks has focused on building gateway-based wireless access networks. With a surge in new devices in the handheld market, presence of multiple NICs in end-users' wireless devices is a reality [6]. This presents the opportunity to explore the application of multi-radio devices in completely ad hoc environments, like a disaster recovery environment or a hospital environment where multiple devices need to communicate with each other simultaneously. In this age of hi-tech hospitals, a doctor/nurse may need to maintain simultaneous communication paths to several devices at the same time for patient monitoring as well as sending/receiving alerts. Unlike the gateway-based scenario for enterprise backbones, this poses a different problem scenario where tree-based architecture used earlier may not apply.

The typical problems of designing a multi-radio network are all present in this new scenario: (a) assigning channels to multiple NICs in a device so that there is minimal interference across hops, and (b) selection of routes to maximize the throughput. Additionally, when there is a limitation on the number of NICs on a device and the number of flows is large, it introduces the problem of timely switching of channel on an interface to prevent starvation of a flow through a node. In this paper, we mainly focus on designing a distributed channel assignment scheme that takes traffic load on a node as the key metric to assign channels to the multiple radios. The routes are chosen based on routes discovered by a well-known ad hoc routing protocol AODV [8]. We show the improvements gained through multiple radios and intelligent channel assignment even when using a standard routing protocol. The starvation of flows is prevented by devising an interface channel switching mechanism. We incorporated interface switching in GloMoSim [3] network simulator.

The rest of the paper is organized as follows. Section 2 provides a brief related work on the subject. Section 3 provides a modeling of a multi-channel multi-radio network and discusses how network is being initialized and maintained. Section 4 describes the algorithms for the multi-radio ad hoc network, with a focus on the channel assignment scheme that we have designed. Section 5 discusses the performance of our algorithms on a 7×7 grid topology with randomly chosen flows. Section 6 concludes with a discussion of the future work.

II. Related Work

Channel assignment problem for multi-channel mesh networks has attracted considerable attention of the research community. It has been proved that optimal channel assignment is a NP-hard problem [11]. Several heuristics has been proposed with a typical architecture of the network in context. For example, Raniwala et al. proposed a centralized load-aware channel assignment and routing scheme for a gateway-based mesh network, that uses a multiple spanning tree based load balancing algorithm adaptive to changing traffic loads [11]. Kyasanur et al. studies the multi-radio mesh network under the assumption of a hybrid setup, where a set of interfaces in a node can switch channels dynamically to establish communication with its neighbors, while the rest of the interfaces are bound to a specific channel [4]. They present a distributed interface assignment strategy that includes the cost of interface switching but is independent of traffic characteristics.

On the other hand, Kodialam et al. has drawn the theoretical bounds on achievable capacity for a multi-radio multi-channel wireless mesh network using a primal-dual approach [7]. Similarly, [9] mathematically formulate the joint channel assignment and routing problem, taking into account the interference constraints, number of channels and radios. They show that the performance of their algorithm is within a constant factor of that of any optimal algorithm for joint channel assignment and routing problem. Unlike majority of these works which solves the channel assignment problem in the context of a gateway-based wireless mesh networks or centralized networks where one node provides required information to other nodes, we propose a purely ad hoc environment. We are proposing a distributed solution to the channel assignment problem in the context of a multi-radio multi-hop ad hoc network without any specific structure.

III. Network Model and Initialization

In this section, we present the network model for the multi-radio ad hoc network. We will introduce the definitions of terms that bring out the important concepts useful in the later sections.

We consider a fixed multi-hop wireless network with n nodes. We represent the network with a directed graph $G = (V; E)$ where V represents the set of nodes in the network, E the set of directed links that can carry data (data links). We are given f source-destination pairs with desired flows between source and destination. We will be utilizing only local topology and local traffic load information to perform channel assignment as used in [10]. This information is collected from a $(k + 1)$-hop neighborhood, where k is the ratio between the interference and communication ranges, and is typically between 2 and 3. We have assumed that all the interfering nodes to a particular node lies in its $(k + 1)$-hop neighborhood.

Definition 1 *$\forall\ v \in V$, p(v) denotes the priority of the node.*

A. Heuristics to calculate $p(v)$

Priority of the node will be an important measure in our channel-assignment algorithm. We present some parameters from which the priority of node is calculated:-

- *Traffic* - Priority of the node increases with increasing Load on the node.
- *Number of Routes* - Priority of node is directly proportional to the number of routes passing through the node.

B. Network Initialization and Topology Discovery

Initially, there are f $s-t$ pairs. First AODV will calculate the routes to follow and then Interface-assignment algorithm will do the assignment. After the routes are known, every node calculates its priority by using any one of the heuristic discussed in section IIIA. And each node will multicast its priority to its $(k+1)$-hop neighborhood. This $(k+1)$-hop neighborhood discovery is simple, as in a sense, node is broadcasting the priority message with a hopcount variable in it. As the neighboring nodes receive this message, if hopcount is less than $(k+1)$ then they will send the message to their neighbors with hopcount increased by 1. If a neighboring node receives the message with hopcount of $(k+1)$ then they will not send the message further. By this process, each node will know the highest priority node in its $(k+1)$-hop neighborhood. This priority is used in the formulation of channel-assignment algorithm. We need to do this priority calculation on a regular basis as the loads on the node can change in course of time.

If a new node joins the network, then it will have to wait till it starts receiving the priority messages from other nodes that means the network is in priority calculation phase, which will be done regularly, and the new node can start sending its priority now.

IV. Channel Assignment and Routing Heuristic

As pointed out in Network Initialization section, every node knows its relative position in its $(k+1)$-hop neighborhood. Each node in the network maintains two list of channels, L_r contains channels that are blocked for receiving and L_t contains channels that can not be used for transmitting. The node with the highest priority in its $(k+1)$-hop neighborhood starts

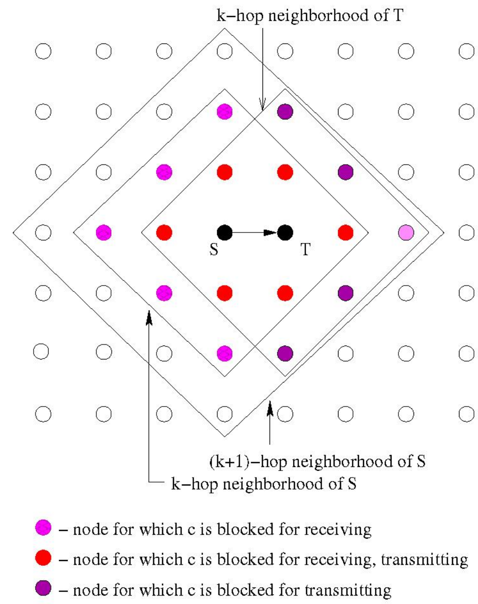

Figure 1: S communicates to T using channel c, $k = 2$.

the Channel Assignment procedure. Let this node be represented by S. S choose the node to communicate by looking at the routing information it has. Let the node chosen by S for communication be T. Then S will select a free channel(if any) for transmitting and check that this channel is also free for receiving at T. For this checking to be done, S transmits a *CHECK* containing channel information packet to T and if channel is free on T, i.e. channel does not belong to L_r of T, then T sends an *OK* packet back to S otherwise it sends a *REJECT* packet specifying that it can not communicate on this channel. That is, channel should not be present in the S's L_t and T's L_r.

After finding such a channel, S multicasts the channel usage information to its k-hop neighborhood requesting the nodes to update their list of blocked channel for receiving. Similarly, T multicasts its channel usage information to k-hop neighborhood by sending a particular packet requesting the nodes to update their list of blocked channel for transmitting as shown in Fig 1. S and T sets one of their interfaces on the selected channel. If no such channel is available which satisfies above stated constraints, then S and T chooses the least loaded channel for communication. Then S will choose another node to communicate from the routing information and start the same process. After assigning interfaces to all the routes passing through S, it will multicast a *RELEASE* packet to its $(k+1)$-hop neighborhood specifying that it has completed its channel

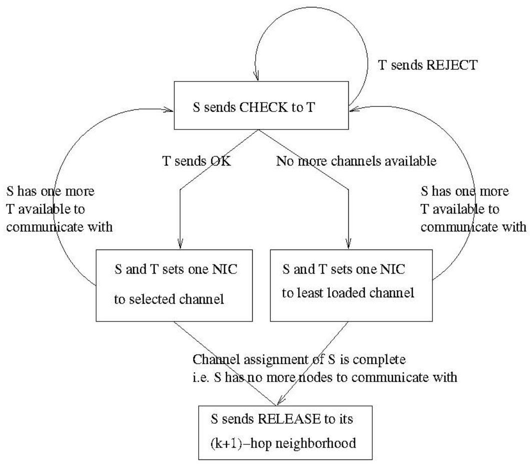

Figure 2: Flowchart of packets exchanged when S does its channel-assignment.

assignment as shown in Fig 2. Now, the next highest priority node in every $(k+1)$-hop neighborhood will start the channel assignment. For all the control packets as shown in Fig 2, there is a reserved control channel which can be later used for data communication after channel assignment is complete.

When every node in the $(k+1)$-hop neighborhood of a node has finished the channel assignment i.e. the node has received the *RELEASE* packet from each node in its $(k+1)$-hop neighborhood then it can start sending the original data. Given algorithm only uses the local information and no prior information about the network topology. Our new algorithm is quite suitable for a dynamic self-starting network.

Lemma 1 $\forall\ v \in V$, *v will start channel-assignment when it is the node with highest p(v) value in its* $(k+1)$*-hop neighborhood which has not done the channel assignment yet.*

A. Interface Switching

In previous subsection, we discussed the basic framework of our channel assignment algorithm. But when the number of interfaces at each node is smaller than the number of channels available, interfaces may have to be switched between channels. In Fig 3, T_p, the priority-shift time, is the interval after which priorities are recalculated and channel assignment is again done. T_i denotes the interface switching interval, that is, how frequently are we switching the interfaces. Except calculating priorities, Channel Assignment algorithm discussed before is to repeated after T_i interval. This mechanism does not require a centralized machine, since in every $(k+1)$-hop neighborhood, node with the highest priority start it.

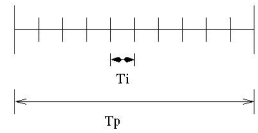

Figure 3: Priority-shift and Interface Switching Time.

Switching an interface from one channel to another incurs a switching delay which is of the order of hundreds of microseconds. So we have to intelligently fix T_i and T_p. Value of T_p should depend upon how frequently the load is varying in network. In the worst case, we need to calculate priorities each time we do interface switching, i.e. $T_i = T_p$ otherwise $T_i < T_p$. However, value of T_i is critical in the performance of our algorithm as too small T_i may lead to high switching delay & no throughput and large values of T_i may lead to the starvation of some flows and high end-to-end delays. Now we discuss the simple algorithm which we use for interface switching.

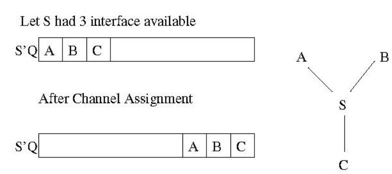

Figure 4: At the start of a T_i, S performs an iteration of switching algorithm.

Each node maintains a Q of nodes to which has to communicate. In the interface -assignment algorithm whenever it get its turn according to its priority, it dequeues one node from Q and assign one interface to the dequeued node and enqueue this node at the rear of Q. If the number of interfaces available on the node is less than the number of nodes it has to communicate with, then it can not do communication with all the nodes in Q simultaneously. After each T_i, node will be able to communicate with the number of nodes present in front of Q equal to the interfaces on the node. Node dequeues these nodes from front of Q and enqueues these nodes at rear of Q, so that the nodes which are starving now should get turn in subsequent T_i intervals. One iteration of the switching algorithm on node S is shown in Fig 4.

B. Routing Heuristic

As most of the routing protocols used today try to get shortest routes. But for a multi-channel multi-radio scenario, shortest path may not be optimal [12]. Instead of choosing shortest paths, routing algorithm should be such that:

- Route should be chosen such that the degree of nodes in the route is less than the number of NICs on the node. This will save a lot of NIC-switching delay.
- Route should be chosen so that it contains the maximum number of edge-disjoint nodes.

So route should be chosen such that it maximizes these two metrics combinedly. This type of routing protocol will also support our interface assignment strategy. For evaluation, current scheme uses AODV for route selection, but this can be improved using the heuristic with the stated goals, which will be part of our future work.

V. Performance Evaluation

In this section, we present results of our performance evaluation based on simulations. We have used GloMoSim [3] for our simulation purposes. We also implemented interface switching in GloMoSim which uses our switching algorithm. We consider a 7×7 grid topology with 16 flows. Source-destination pair for each flow are calculated randomly and each flow has a fixed demand equal to the bandwidth of single link so that the network saturates. We vary the number of radios from 2 to 3, and the number of channels from 1 to 8. Every channel has same fixed bandwidth associated with it. We have taken the interference to communication distance ratio to be 2, i.e. $k = 2$. Grid unit distance (between any two consecutive nodes) is such that the two consecutive nodes are always in communication range and nodes 2-hop away are in interference range, and the nodes that are 3 or more hop away are not in interference range as shown in Fig 5.

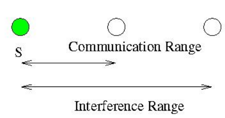

Figure 5: Interference and communication ranges for S, $k = 2$.

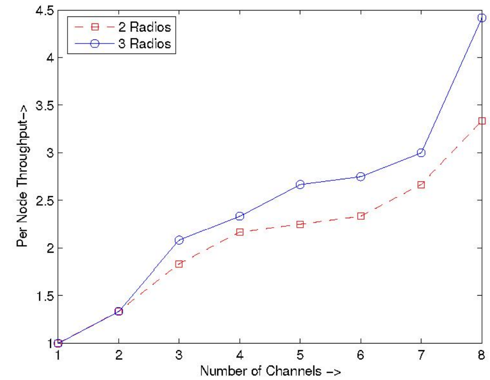

Figure 6: Throughput improvement with increasing Number of Channels

Fig 6 shows the per-node throughput improvements as the number of channels are increased. Per node throughput increases quickly as we move from channel 1 to 3 and then 5 to 8. In between 3 to 5, throughput increases linearly. Throughput is normalized with the single channel single radio throughput. These throughputs are the throughputs with T_i of 10 seconds and T_p of time ∞, i.e. there is no priority recalculation as the loads in the network are not time-varying.

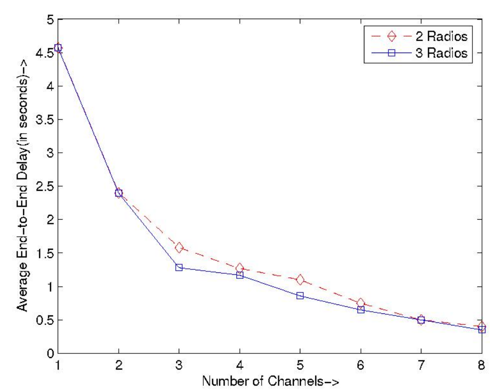

Figure 7: Average End-to-End Delay versus Number of Channels

Fig 7 shows Average End-to-End delay versus the number of channels. As expected, on increasing the number of channels delay decreases swiftly and the response time also decreases. But as the number of channels are increased from more than 3, the decrease is not large but there is still a decrement.

Fig 8 depicts the important behavior of the parameter T_i on per node throughput. These simulations were run for a period for 60 seconds. We normalize

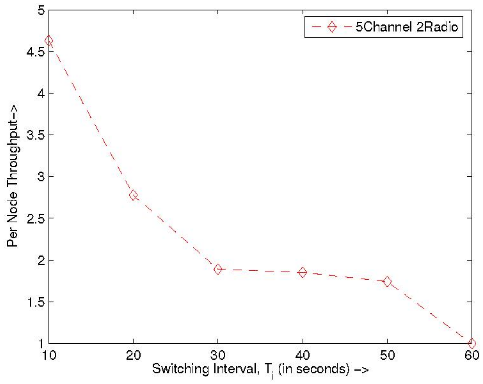

Figure 8: Per Node Throughput versus T_i

the throughput with no switching throughput. Since, GloMoSim does not incorporates the interface switching delay, we were not able to get optimal T_i, but we got that interface switching increase the throughput in a considerable amount. High improvement justifies the intuition behind our switching algorithm.

Parameter T_p, the priority-shift time, can be tuned on the same frequency as how the traffic is varying in the network. As priority of a node determines how much traffic does it have, so we need to calculate priorities repetitively only if the network has time-varying traffic.

VI. Conclusions

The recent developments in multi-radio devices has encouraged research in architecting multi-radio multi-hop networks. Looking at the growing number of multi-radio end-user devices, we envision a future of multi-radio multi-hop ad hoc network with no inherent structure. In this paper, we present a distributed channel assignment scheme for such a network using a priority tagging mechanism for each node. The priority determination for a node is dependent on the traffic load it carries. Currently our route selection uses the well-known AODV routing protocol.

However, we identify possible improvements for AODV in a multi-radio scenario that we plan to investigate in future. The efficiency of our channel assignment algorithm is demonstrated through simulations on GloMoSim. We have shown that per node throughput improves significantly when we increase the number of channels and radios available per node. We have also given a simple channel switching strategy which saves the network from arbitrary starvation. We incorporated our interface switching strategy in GloMoSim and showed that it also increases the throughput as it tries to distribute the channels between flows regularly.

References

[1] IEEE Std. 802.11-1999 "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", 1999.

[2] IEEE 802.16, Air Interface for Fixed Broadband Wireless Access Systems.

[3] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. GlomoSim: A Scalable Network Simulation Environment. Technical ReportCSD Technical Report, #990027, UCLA, 1997.

[4] P.Kyasanur, N.H.vaidya, "Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks", IEEE WCNC 2005.

[5] Atul Adya; Paramvir Bahl; Jitendra Padhye; Alec Wolman; Lidong Zhou, "A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks", ACM Mobicom 2003.

[6] Nokia N91, http://www.nokian91.org/.

[7] Murali Kodialam, Thyaga Nandagopal, "Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks", ACM Mobicom 2005.

[8] Charles E. Perkins, Elizabeth M. Royer"Ad-hoc On-Demand Distance Vector Routing", 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999.

[9] Mansoor Alicherry, Randeep Bhatia, Li Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multiradio Wireless Mesh Networks", ACM Mobicom 2005.

[10] Ashish Raniwala, Tzi-cker Chiueh, "Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network", IEEE Infocom 2005.

[11] Ashish Raniwala, Kartik Gopalan, Tzi-cker Chiueh, "Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks", ACM SIGMOBILE 2004.

[12] Richard Draves, Jitendra Padhye and Brian Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks", ACM Mobicom, 2004.