Architecture for Multi-Interface Multi-Channel Wireless Networks

Mridul Barsaiyan
Y3184
mridubar@iitk.ac.in

Project Supervisor:
Dr. Dheeraj Sanghi
dheeraj@cse.iitk.ac.in
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

Abstract

With the increase of usage of wireless networks for purposes where the nodes are either stationary or minimally mobile, focus is also on increasing the network capacity of wireless networks. One such way is to use non-overlapping multiple channels provided by 802.11 by using multiple interfaces per node. Multiple non-overlapped channels exist in the 2.4GHz and 5GHz spectrum. However, most IEEE 802.11-based multi-hop ad hoc networks today use only a single channel. As a result, these networks rarely can fully exploit the aggregate bandwidth available in the radio spectrum provisioned by the standards. Under this scenario, several challenges need to be addressed before all the available channels can be fully utilized. Major ones are design of underlying routing protocols and channel assignment algorithms to go with the existing off-the-shelf hardware. This report summarizes the existing work done in this direction. Also, it lists out three simulators which might be used for simulation studies of such architectures. Lastly, it discusses an extension of NS-2.29 compatible with latest version of gcc and supporting multiple-channel mutiple-interface assignment. Also it gives simulation study of implementation of Hyacinth[2] for more than 2 network interfaces per node.

1. Introduction

Wireless network technology, despite being extremely useful in mobile communication and computing application, suffers from low link-layer data rates. The 54 Mbps peak link-layer data rate of IEEE 802.11a/g wireless LAN interface stands no chance in front of huge bandwidths provided by wired technologies. Moreover, overheads of packet loss, packet errors, contention and packet headers, drastically reduce the actual goodput available to the wireless network applications. The data rate also falls quickly with increasing distance between signal source and destination. Interference from adjacent hops in a multi-hop network further decreases the available bandwidth. Usage of multiple channels thus removes both of the problems - it extends the available bandwidth and removes the problem of interference as now simultaneous communication is established between adjacent hops on a non-overlapping channel.
The sudden increase in the rise of interest in this research area can be owed to wireless hardware costs which are steeply falling. Hence, it is now feasible to equip nodes with multiple 802.11 wireless interfaces. As we cannot equip the nodes with an interface for every non-overlapping channel due to power consumption and size constraints, we need to devise channel assignment algorithms which would switch the interfaces from one channel to other albeit at the cost of *switching delay*. Also, we need hardware that would

allow functioning of more than one interfaces at the same time to enable simultaneous communication on different channels. The decrease in prices has led to the emergence of cost-effective wireless applications and systems. Though the deployment by large has been limited to small enterprises and home environments, efforts are now towards usage of wireless technology in dense networks to serve more users over a wide area.

Multi-radio and multi-channel architecture has special usage is wireless mesh networks (WMN)[10]. In a mesh network, nodes act as repeaters to transmit data from nearby nodes to distant nodes in the network. Their special utility is in case of providing an inexpensive last mile broadband internet connectivity. In some cases the mesh may be serving as an extension to a wired backbone, thus decreasing the need of a dense physical wire network and hence the cost of maintenance.

WMNs, though similar in concept, differ on some points with adhoc networks. The nodes in WMNs, despite of being enabled with wireless capacity are stationary in nature or show limited mobility. Thus, topology of WMNs are more or less static. Also, the nature of data flow in a WMN is also the same. For example in an enterprise, traffic mostly consists of internet usage by the nodes or communication between nodes for local file transfers. This provides an opportunity for optimisation of WMN's for certain traffic characteristics to improve efficiency. The traffic distribution in a WMN is also in fixed direction, generally to/from a wired/wireless backbone.

Introduction of mesh networks also calls for efforts in the direction of increase in network capacity. Usage of mesh networks as an extension to a wired backbone will lead to surge in bandwidth-intensive applications like video-sharing. Usage of multiple channels available in IEEE 802.11a/g standards offers a promising avenue in this regard. the IEEE 802.11b/g standards and IEEE 802.11a standard provide 3 and 12 non-overlapped frequency channels, respectively. Utilization of these multiple channels effectively would increase the bandwidth substantially. Efforts have been made in this direction in two directions:

1. Modifying the MAC protocols to support multi-channel networks. The research focuses on finding an optimal channel for packet transmission enabling multiple parallel transmissions in the neighbourhood by avoiding interference.
2. Modifying system software i.e. making changes in other network layers so that benefits of multiple channels can be used by using off-the-shelf commodity 802.11 interfaces.

Here we discuss the efforts made in the second direction, as it makes deployment of such networks for large enterprises more feasible.

2. Issues and Challenges

Currently available off-the-shelf 802.11 interfaces use one channel at a time, with the ability to switch between channels. Thus the ability of a node to simultaneously transmit over different channels is decided by the number of Network Interfaces (NICs) on the host. The basic assumption is that the number of interfaces on one node is generally less than the number of available no-interfering channels. This is generally valid as the number of such channels is upto 12, whereas the number of NICs on a host might be limited to 2-4 due to power consumption and size constraints. Also, there might be channels which overlap partially. Such channels, in case of simultaneous use might cause interference problems. Hence, for simplicity most protocols assume the use of completely non-overlapping set of channels. *The challenge is the efficient and effective use of fewer number of NICs on a host to utilize all available non-overlapping channels to increase network capacity*.

2.1 Channel Assignment

To utilize a greater number of channels with fewer NICs per node we need the ability to switch a NIC between channels. For example, if we have a single NIC per node in a network of four nodes, and each node is fixed on the same channel, the opportunity to use the other available channels is wasted. We cannot fix the NIC to other channel, as then the nodes on two different channels will be unable to communicate with each other. Thus, we need a channel assignment algorithm which coordinates between the nodes and schedules assignment, and if necessary, switching of channels among the NICs to utilize multiple channels. For maximum benefit, such a channel assignment algorithm needs to adhere to certain demands of the network. There are a number of ways of approaching the channel-assignment problem each with some issues[6]:

Fixed channel assignment

In this approach we come to a optimal assignment of channels to the available NICs on the nodes and keep them fixed always. One such way would be to use as many channels as the number of NICs available per node i.e. if we have *n* number of NICs on a node we fix them to *n* different channels. This would lead to a very simplified scenario as we will just need to manage simultaneous communications on multiple channels. This is however not a optimal solution as it will leave many channels unused and be inefficient, specially in the case where the number of NICs is very less as compared to the number of channels.
Another approach would be to fix NICs on different nodes to non-disjoint sets of different channels. Thus each node will share atleast one different channel with other nodes. This would balance the load on different channels and utilize all channels effectively. Though, this might simplify the protocol, a simple assignment of channels might hamper network connectivity. It might cause network partitioning i.e. disjoint sets of nodes which have no connectivity between them (Fig. 1), even if nodes have multiple NICs.

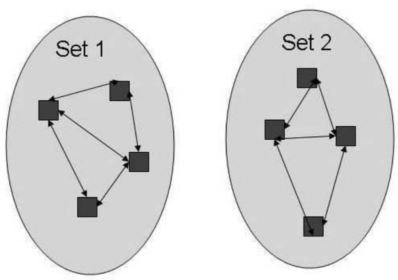

Figure 1. Fixed channel assignment leading to disjoint sets

Dynamic channel assignment

Alternate way of channel assignments is to frequently switch channels for different interfaces. Through this flexible approach, each node balances its load over all channels in due course of time. However, this policy will require frequent channel switching leading to switch delays. Also it requires careful co-ordination between nodes to ensure that two nodes which need to communicate have alteast one of their interfaces use a common channel.

Hybrid assignment

In a scenario where we have more than one NIC on each node, an intermediate path is also available. We can fix one NIC of each node to one common channel which would server as a control channel to coordinate channel switching of remaining NICs between nodes. Thus we have one fixed channel interface and rest dynamic channel interfaces.
Fine-grained channel coordination is also necessary with different channel-assignment techniques. In case of single NIC per node, to ensure utilization of all channels the NIC would be switched frequently. However, to ensure connectivity we must ensure periodic synchronised switching to a common channel. Another possibility is a routine of channel switching to be followed by each node which would enable to other nodes to know when it would share a common channel with the node. Such channel coordination will also provide the network with the capability of local broadcast which is necessary for passing of important network information such as routing data. Thus the constraints that need to be handled in channel assignment algorithms can be summarised as follows

1. The number of channels that can be effectively utilized depends on the number of NICs available per node

2. Neighbouring nodes should be bound to a common channel or be able to switch onto a common channel when required so

3. The expected load on a channel should not increase the capacity of the link

4. The total number of available channels is also fixed

2.2. Routing

Existing routing protocols, though support multiple interfaces at each node, are not suitable for multi-channel environment. Routing protocols like AODV and DSR typically select shortest-distance routes. However, in multi-channel networks switching delays and distribution of routes plays an important factor. The shortest route in a multi-channel network might include multiple channel-switchings which would cause further delay in transmission than a longer route which requires much lesser number of channel switches. In case of multiple NICs, a multi-channel routing protocol needs to distribute the routes so as to enable maximum utilisation of multiple channels simultaneously. Thus, we need different routing metrics for taking routing decisions in a multi-channel and multi-interface network than conventional metrics.

2.3 Other Issues

Other than the issues discussed above there are some more that are important to the design issues of multi-channel multi-interface networks. Mobility has not been taken into consideration in most of the algorithms proposed in the available architecture. Mobility leads to frequent change in the neighbour set of a node which in turn leads to change in load-balancing of routing protocols and channel interference. Therefore, in case of mobility we need more frequent exchange of information between nodes causing overhead. Also mobility might lead to channel fading leading link breakage.
Another issue is using power control approaches for topology control. In a dense network, multi-channel networks are beneficial as due to diversity in channels the number of contending transmissions become small minimising delays. In case of low network density, frequent interface switching will ensure connectivity. A right combination of channel assignment and routing algorithm will enable the architecture to dynamically adapt to the density of the network.

3. Channel Assignment algorithms

Much of the work on channel assignment algorithm for multi-channel multi-interface networks can be accredited to Prof. Vaidya et. al. Here is a discussion of key channel assignment algorithms proposed in literature.

3.0.1 Neighbour partitioning technique [2]

One algorithm of channel assignment uses partitioning of neighbours into groups depending on the number of NICs in a node. Suppose each node in a network has *r* nodes. Then we divide the set of neighbours of the node into *r* partitions and assign one NIC to each group. Next, we take up each node and continue with partitioning of neighbours taking into constraint previous partitions. The process is repeated until all nodes have partitioned their nodes. Each group is then assigned a channel least-used in the neighbourhood. This algorithm allows the usage of more channels than the number of interfaces on each node. However, the algorithm is more successful in network topologies that have equal load on each virtual link. In case of skewed traffic loads (as in the case of WMNs with wired backbone where the traffic is to/from a certain node), the algorithm will not perform well.

3.0.2 Mutli-channel MAC [8]

MMAC is a link-layer protocol for single interface networks to utilize multiple channels. MMAC requires time synchronisation between nodes to coordinate channel assignment. In MMAC time is divided into quantum units called *beacons*. At the start of this beacon, each node is forced to assign a common channel to its NIC. This ensures connectivity between nodes and a chance for all nodes to exchange information for further communication.
As soon as all nodes are on a common channel after a random interval (to avoid contention) the node which has data to transmit (SRC1) sends a ATIM packet to the destination (DST1). The ATIM packet includes channel information about the usage of channels in SRC1's neighbourhood, termed as the *Preferred Channel List (PCL)*. On receiving this ATIM packet, DST1 matches its own PCL and decides the least used channel as communication medium and replies with an ATIM-ACK packet containing the above information. The transaction is confirmed by SRC1 by sending an ATIM-RES packet. Meanwhile, the neighbours of SRC1 and DST1 update their PCL's

on hearing the conversation. Similarly, other SRC-DST pairs get ready for communication within the stipulated ATIM window. After the ATIM window, communication as decided earlier continues and data is sent by SRC1 to DST1 on the pre-decided channel. Simultaneously, other SRC-DST pairs do data transaction. The communication stops as the beacon interval approaches and all the nodes switch back to the common channel. Thus, MMAC enables equal load-distribution on all channels as well as adequate opportunities for broadcast during the ATIM window.

3.1 HMCP: Hybrid Multi-Channel Protocol [8]

This proposed protocol assumes atleast two NICs per node. This algorithm works by dividing the NICs on a node in two sets - fixed and switchable. While the fixed NIC is assigned to a particular channel for a relatively much longer period of time, the switchable NICs switch to different orthogonal channels as required. Each node maintains a *Channel Usage List (CUL)* which keeps account of the channel being used by the fixed NIC of all nodes. When a node needs to transmit data to some other node, it switches its switchable NIC to the fixed channel of the desired node and starts a conversation.

Fixed channel assignment

The assignment of the channel to be assigned to the fixed NIC of a particular node takes place in the following steps

1. Initially, each channel chooses at random a certain channel as its fixed channel and update their own *CUL*.

2. Periodically, every node broadcasts on each channel its currently fixed channel. All other nodes hear this broadcast and update their CUL's

3. Each node periodically consults its CUL. If its fixed channel is considered to be too crowded it has probability p to change its fixed channel to a less crowded one. If it decided to change its channel it broadcasts the information on all channels so that other nodes update their CULs. The probability p prevents the occurence of an event where every node on the crowded channel list decide to change its channel to a less crowded one, crowding another channel in turn. this process is repeated until all channel are equally balanced in terms of number of nodes which use it as a fixed channel.

3.2 Load Aware Centralized Channel Assignment

According to this algorithm, channels are assigned to multiple interfaces on a node centrally according to the metric, *degree of interference* experienced by the link. *Degree of interference* is defined as the sum of expected load that a link may experience on a particular channel in its interference region. The algorithm requires link load as input. The channel assignment takes place in the following steps:

1. The links are sorted in descending order of their respective loads. A table which keeps record of degree of interference of channels for each link is maintained.

2. The link is assigned to the channel with the lowest degree of interference and one interface of both nodes in the link are bound to that channel

3. This process is repeated until we come to a stage where all interfaces of a node have already been bound to a channel. In this case, if the two nodes in the link have a common channel, the link is assigned to that channel. Else, the degree of interference assuming various possible changes is measured and the link is assigned to the channel that that shows minimum degree. This process is carried out till we reach equilibrium and then onwards we have a static channel assignment.

Though the algorithm is simple, it requires initial expected load on all the links which might not be feasible. Also, as it is a centralized channel assignment, network topology needs to be known beforehand. Thus, the algorithm gives best results for static network topology and static network loads.

4. Routing Protocols

Although most routing protocols for single channel assignment would work with proposed channel assign-

ment algorithms, they will not be optimal. In multiple channel network, as opposed to single channel network, shortest-path metric is not optimal. Hence, the challenge is to suggest the best suitable metric for such networks. While deciding the metric for such a case, we need to keep in mind the switching delay, channel diversity and the conventional resource usage i.e. the number of hops.
When we choose a route for a packet in a multiple channel network, though it might be shorter than their paths, it might include channel switching for a majority of hops on its way. Switching of interfaces to different channels incurs switching delay which should be minimised. At the same time a node should transmit and receive on different channels as this enables it to do so simultaneously, increasing the throughput. Hence, a route where all nodes receive and transmit on different channels should be preferred. However, total hops along the route also have a weightage as there should not be inefficient use of resources along the route. A routing strategy for multi-channel environment should support these factors.

4.1 Multi-Radio Link Quality Source Routing [9]

MR-LQSR is a LQSR protocol with a different weight metric, Weighted Cumulative Expected Transmission Time (WCETT), suitable for multi-channel networks. It is basically a link-state protocol derived from DSR. In DSR, the protocol assigns equal weight to all links and comes with a good path by simply adding the path lengths for the destination. This is where the routing protocol for multi-radio network differs. It uses WCETT as the path metric, which differentiates between routes not just on the basis of length. MR-LQSR is designed for networks consisting of nodes with restricted mobility, having one or more wireless radios tuned to different non-interfering channels which do not change very frequently. The metric takes the loss rate and the bandwidth of a link into account before assigning weight to the link. Also, addition of a node to a path always increases the cost of the metric. This is to ensure that no extra nodes are added unnecessarily to the route, as extra nodes consume more resources and introduce extra delay. To make it more suitable for multiple-channel environment, the metric should take into account throughput loss due to interference among links operating on same channel. Hence a path where channel changes with hops is better than a path with same channel throughput.

4.2 Multi-Channel Routing [7]

MCR is based on Multi-Channel routing metric which is a modified version of WCETT and includes switching costs. considering switching costs during route selection ensures that a path with frequent switching over channels is not preferred. MCR is calculated by adding another factor in WCETT, the probabilistic switching delay on the path. This factor is obtained by measuring the probability of a packet being on a path being switched multiplied by the switching delay i.e. the latency in switching of channels.

5 Hyacinth [10]

While we have discussed some channel assignment algorithms and routing protocols independently, Hyacinth is a complete architecture for multi-channel multi-interface wireless networks. The architecture, with routing and channel assignment as major design issues, works directly with commodity 802.11 interfaces with only system software modifications. The architecure is proposed for a WMN where there are some wireless gateways which are connected to the internet backbone with physical wires, while the rest of the nodes form a multi-hop WMN. The routing protocol separates out the load on each channel and uses the information to balance out loads on other links. Bandwidth available to each node is computed taking into account the load on interfering nodes. In the target architecture all infrastructure resources are located in the wired backbone and access points provide connectivity to these resources to the end-user by forming an ad hoc network among themselves and access them through the wireless gateways.

Each node is equipped with multiple NICs, with a channel assigned to the NICs for a relatively very long time. For direct communication between nodes they should be in the communication range of each other and share a common channel. Nodes that use the same channel interfere with each other, even though not communicating with each other.

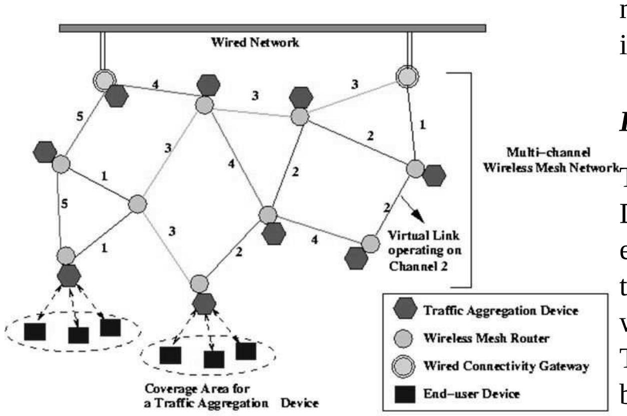

Figure 2. Hyacinth network topology *http://www.ecsl.cs.sunysb.edu/multichannel/

5.1 Channel assignment algorithm

As in the proposed topology we need to ensure connectivity with the wireless gateways, we cannot simple assign the least used channel. Each node needs to share atleast one common channel with all of its neighbours, whereas to avoid interference it should share a common channel with the fewest number of neighbours possible. The channel asignemnt algorithm in Hyacinth is distributed and load-aware. The assignment can be divided into

Neighbour-Interface binding

A distributed channel assignment is heavily channel dependent. A local channel re-assignment decision might lead to major changes in the overall network channel distribution. As a remedy, due to its topology, Hyacinth architecture divides its NICs into two disjoint sets: UP-NICs which the node uses to communicate with its parent node and DOWN-NICs which it uses to communicate with the children nodes. A node is responsible for assigning channel to its DOWN-NICs. Each node's UP-NICs is associated with a parent's unique DOWN-NIC and is assigned the same channel. A local channel re-assignment decision will not affect the channel distribution of UP-NICs of a node. This way, we can apply a channel assignment methodology going from top to bottom, without caring about the consequences on other nodes.

Interface-channel assignment

The node is responsible for channel assignment to its DOWN-NICs. It does so by estimating the load on each channel in its interference neighbourhood. For this, the node exchanges channel usage information with its children in the interference neighbourhood. The sum of loads contributed by all interfering neighbours that use this channel is the aggregate traffic load of a particular channel. Based on information from its parents and children, the node assigns the least used channel to its DOWN-NICs, which in turn assigns the same channel to corresponding UP-NICs of the corresponding children. The nodes higher up the topology get more priority while choosing their channel.

5.2 Routing

As already said, the hyacinth topology is such that the major traffic is to/from the wired backbone. Thus, the objective of the routing protocol is to find the best route to the wired network for each node. Each wired gateway acts as a root of a spanning tree and each node is part of atleast one spanning tree, while keeps other trees as backup option for failure recovery as mentioned.

The final topology is determined by the cost metric used in the routing. The hop count between the joining node and the gateway at the root of the tree is one of the logical cost metrics. This makes the path to the wired network shortest and leads to rapid convergence of the routing phase, but does not balance load in the network. To take care of the netowrk load, gateway link capacity i.e. the residual uplink bandwidth of the wired gateway at the root of the tree with the wired backbone, is another metric. The link capacity is calculated by subtracting already used bandwidth from the total uplink capacity. However, as the bottleneck in the network can be any link from the node to the root rather than the gateway link, *path capacity* is a more general cost metric. The capacity of wireless link is calculated by subtracting the aggregate usage of the channel in the neighbourhood from the total capacity.

6 Network Simulators

A network simulator is a software that simulates the network without a netwowrk actually being present. Compared to the cost and time involved in setting up an entire test bed containing multiple networked computers, routers, infrastructure, network simulators are relatively fast and inexpensive. They allow testing of scenarios that might be particularly difficult or expensive to simulate using real hardware. Networking simulators are particularly useful in allowing designers to test new networking protocols or changes to existing protocols in a controlled and reproducible environment. There are a wide variety of network simulators, ranging from the very simple to the very complex. Some of teh Network Simulators that were tried are:

6.1 OPNET[4]

In the search of a substitute network simulation software, OPNET IT Guru was seen as a possible choice. OPNET IT Guru is a widely used networking and simulation software. It enables one to create a virtual network consisting of relevant hardware, protocols, and application software. It also has inbuilt tools for dynamically investigating the created network. However, the free academic version is meant for some specific lab exercises for teaching purposes and is limited in its functionality and resources. Another hurdle is the lack of free appropriate tutorials and manuals for the software. Thus, it did not suffice for simulation of multi-channel environments.

6.2 Network Simulator - NS2 [1]

NS2 is an open source simulator targeted for network research. NS2 official release does not support multi-channel wireless network environment. Therefore, special releases are needed for such simulations. There are several issues related with installation of NS. *NS-2.1b9a does not install for gcc version ¿3.2.* While most of the implementations for multiple channel algorithms (Hyacinth, MMAC, TENS) are on NS2.1b9a, it is difficult to make these implementation compatible with latest version of NS i.e. NS-2.29.

NS wireless code

In the NS implementation of wireless networks, the layers are defined as

- Routing - topmost layer
- Link Layer - either queue or send the packet
- ARP - address resolution
- Ifq - queueing
- MAC - MAC parameters and timers
- NetIF & Propagation - physical layer with some power options
- Channel - channel assignment and connectivity

A packet is passed from one layer to another by calling recv() function of the layer UPSTREAM or DOWNSTREAM. The neighbour nodes that will be affected (i.e. a common channel would interfere) from a given node are determined by *WirelessChannel::getAffectedNodes()*. If the nodes share a common channel, copies of packet transmitted by a node will be heard by all neihbouring nodes.

Hyacinth NS implementation

Hyacinth has been implemented as a code in C language which on giving parameters produces tcl scripts for the use of NS-2. The authors have also modified NS-2 version 2.1b9a to support multichannel and tag based routing. The code has been organised as:

- Controller.c : Main function, initialization of the network
- Channel.c : Channel assignment and interference estimation function. Greedy algorithm based on minimum load
- Route.c : Assign routing, Djikstras Algorithm
- Topogen.c : Generates topology based on a random seed between 0 and max-1 (max is the number of nodes)
- Trafficgen.c : Generates traffic between nodes based on random seed

The code on execution with appropriate parameters produces tcl scripts namely graph.tcl, traffic.tcl, monitor.tcl, and channel-route.tcl which simulate Hyacinth on modified NS2.1b9a . NS2.1b9a compiles only on gcc version 2.96 or less.

6.3 TENS [11]

The Enhanced Network Simulator (TeNs), is an extension of Network Simulator (NS-2). TeNs is an attempt to address the deficiencies of *n*s in the modeling of IEEE 802.11 MAC layer protocol, which is highly simplified in the original *ns* . Apart from a more realistic implementation of this, it also incorporates additional features like multiple interfaces support for mobile nodes, a static routing protocol for wireless scenarios, and also features inclusion of simple directional antennas. Multiple interfaces support is what makes TeNs a suitable candidate for simulation of multi-channel multi-interface wireless networks.

7. Results

Now we have an extension of NS-2.29.2 which supports multi-interface multi-channel environments. In the current extension the routing has manual and each node is equipped with 5 interfaces (can be changed by changing #NUM_NICS variable in *mac/mac-802_11.cc* in the ns folder), assigned to different orthogonal channels. This platform can be used for implementation of various routing and channel assignment algorithms proposed in the literatures discussed above. For validation of the multi-interface multi-channel extension, some sample scripts for simple networks were executed. We observe almost a double increase in the throughput in the case when we use multiple channels and multiple interfaces.

For each of the above simulations, traffic of *UDP* packets of 1000 bytes each for 5 seconds was used. In Figure 3, we observe that as we increase the load on the network, the bottleneck, after which packets are being dropped comes much later in the case of multiple networks. For this a simple topology of two pairs of nodes communicating with each other simultaneously, using same channel in first case and different channels in the other. Similarly we see a similar behaviour in case of 3 nodes, with one source, one intermediate node and one receiver (Figure 5) and one source and two receivers (Figure 4). The data from Figure 3 and Figure 4 was then plotted against each other in Figure 6 to verify that in this extension, no difference in performance overhead is observed on usage of multiple interfaces on a single node. It practically behaves like two nodes working simultaneously on two different channels.

Figure 7 shows us an important observation about interference level in NS. The topology used in Figure 3 was used. In this simulation, the packet load was fixed

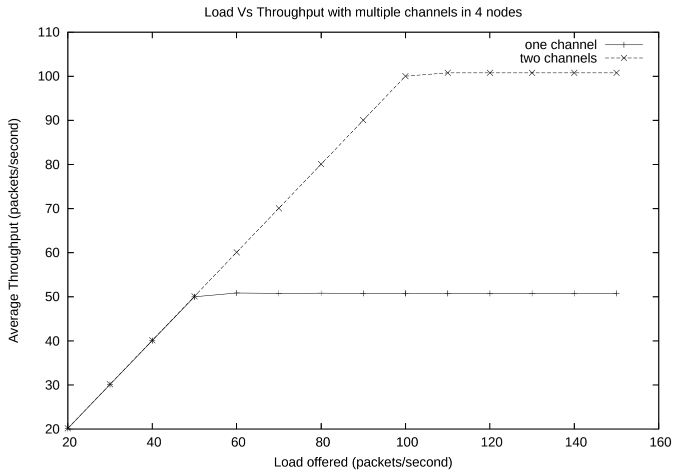

Figure 3. Topology: 4 nodes, pairwise simultaneous communication

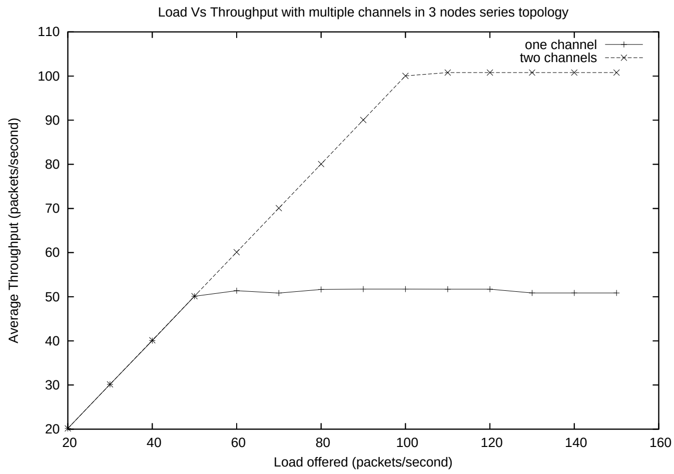

Figure 4. Topology: 3 nodes, one source with two interfaces and two receivers

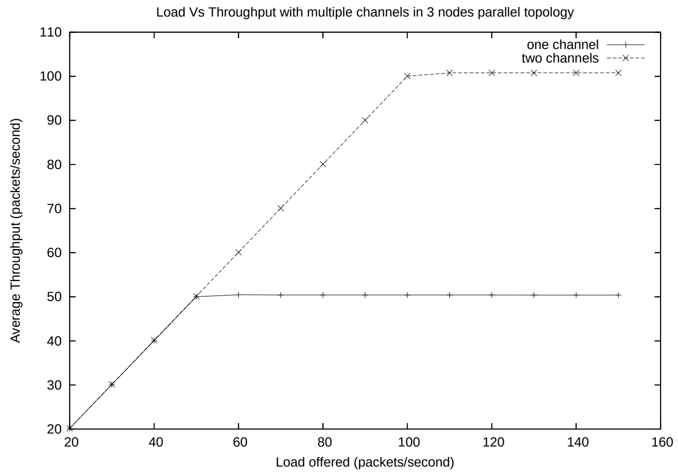

Figure 5. Topology: 3 nodes, one source, one forwarder and one receiver

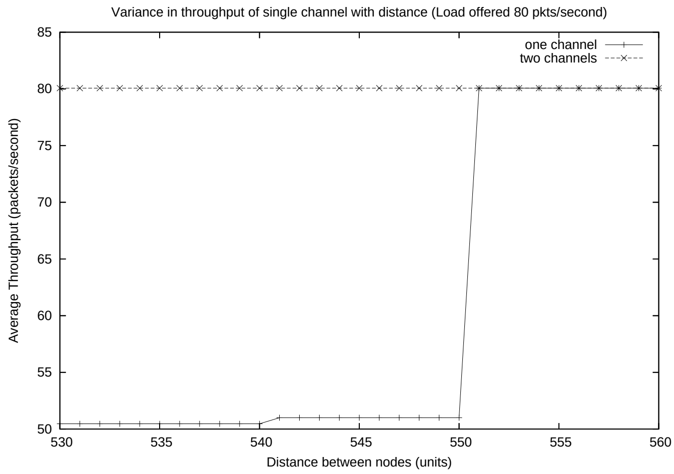

Figure 7. Topology: 4 nodes, pairwise simultaneous communication fixed distance among pairs, increasing distance between pairs

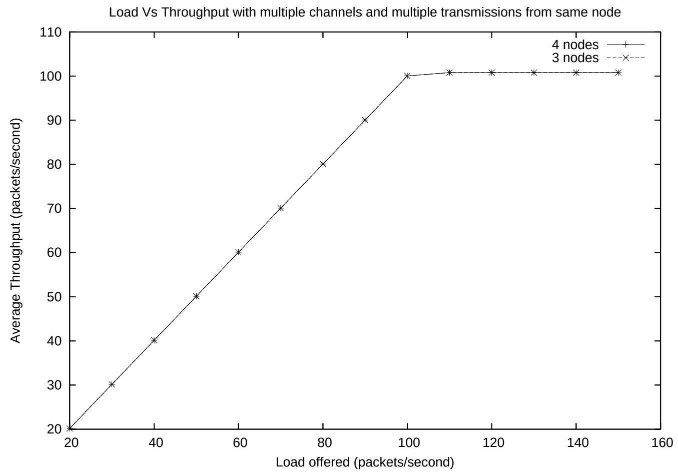

Figure 6. Plotting of multiple channel data of figure 5 and 6

at 80 packets per second (The value was chosen on the basis of results in Figure 3, so that we distinctly observe when the change in performance occurs in the case of single channel scenario). Then the distance between the pairs (distance between the nodes in a pair was kept fixed at 100 units) was increased. We observe a sudden change in performance at a distance of 550 units. WE can infer that the interference between the pairs disappears at this distance and hence the performance is similar to the case of multiple channels. We also observe that the interference is discrete. This distance can be tweaked by changing the power of the antennas in NS simulation. Thus, the above simulation results concur with that proposed in theory and verify the current extension of NS that supports multiple interface and multiple channel wireless network environment.

Further simulations were done with the hyacinth implementation provided by the authors of [2]. The focus of the study was to measure the performance when the number of nics on a node is more than two. The simulations were conducted in dense topologies. For the given graphs a topology of randomly chosen 61 nodes was used in a grid of 9x9. 10 flows of random varied bandwidths between 0 and assigned maximum bandwidth were generated. Figure 8 shows us the variation in the total cross-section goodput as opposed to the increasing input end-to-end bandwidth when the number of interfaces is increased. The change is greater in networks when the traffic is denser i.e when the network becomes saturated.

Multi-Interface Multi-Channel extension

The Multi-Interface Multi-Channel extension of NS-2.29 now enables us to simulate such wireless network environments in NS. Other simulators either do not provide sufficient features or are not freely avail-

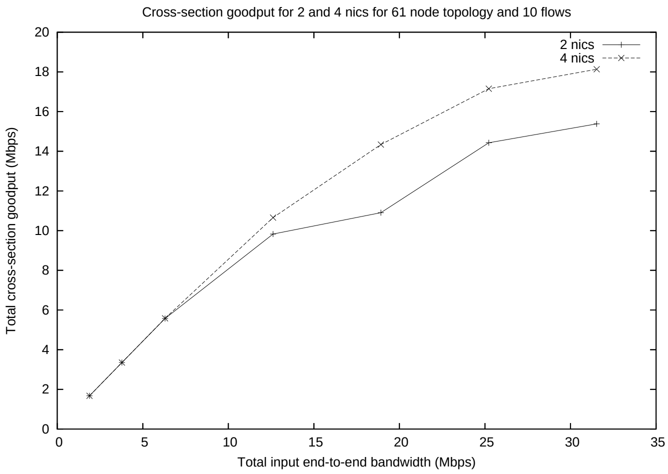

Figure 8. Topology: 61 nodes, 10 traffic flows of random bandwidth

able. The extension can serve as platform for performance evaluation of channel assignment protocols and routing protocols developed for such wireless environments.

The extension equips each node with 5 networks interfaces. Each interface on a node can be bound to a particular non-overlapping channel. There can be any number of channels as the user may require. The extension also supports manual routing as proposed in the Hyacinth implementation on NS-2.1b9a. Hence, to simulate a channel assignment protocol for multi interface environments, We need to generate a script that has manual routing and the interfaces of the nodes are bound to the channels as required by the protocol. Thus, the implementation of the protocol can be carried out in any programming language which in the end generates a tcl script that can be run with the extended version of ns.

First we must generate the topology as required i.e. the placement of the nodes in the network. Next we decide upon the traffic that we need to simulate on the given topology. Given the traffic and topology, the implementation of the protocol should assign different channels to the network interfaces of the nodes and also find routes for the traffic. The routing that is generated using this protocol can then be generated in the form of a script in the form of mnual routing where we hardcode the route that each traffic flow needs to take. Thus, the implementation can be any programming language with the end result as a tcl script. Though the number of interfaces on a node is 5, it can be increased by making changes in the NS code. For protocols that require fewer number of interfaces, the implementation can be done assuming the required number of interfaces and rest of the interfaces can be assigned a fixed channel that is never used. Unless an interface is specifically instructed to be used in a particular route it does not interfere with the performance. The network environment can be altered by changing the network parameters available in NS.

The example tcl script for Hyacinth serves as an excellent example. The implementation is in the form of C programs which randomly generate topology and traffic flows. These are then used for channel assignment according to the protocol and generate the manual routing and channel assignment to the interaces of all nodes. The network parameters are coded in a base script which use the automatically generated scripts for the desired simulation.

8. Conclusion

Multi-interface multi-channel wireless algorithms are useful for exploiting the wasted bandwidth available in the form of orthogonal channels in the IEEE 802.11a/g interface. Special routing protocols and channel assignment algorithms are required for such network environments. Wide usage of such networks would be a great boon to last-mile internet connectivity and development of low-cost wireless mesh networks where only a wired backbone is used as a support. Extensive literature survey in this field shows that though there have been many theoretical advances in the field, there is lack of any real extensive deployment. Most of the implementation on simulators like Ns2 were either out of date or broken. The extension of NS provides a framework on which other channel-assignment and routing algorithms for multi-channel multi-interface networks can be implemented.

Hyacinth is the only real architecture that, though being designed for very specific topology, combines the routing and channel assignment algorithm to provide a complete architecture. The changes in NS to support multiple channel and multiple interface networks is majorly inspired from the changes in the implementation of Hyacinth.

9. Acknowledgement

I would like to thank Dr. Dheeraj Sanghi for his support and guidance on literature work available in the field and also for helping out for acquiring OPNET's professional license, which unfortunately is not given to Indian universities. I would also like to thank Mohit Mundhra for his initial help on literature and unsuccessful attempts at Hyacinth's implementation. Last but not the least, I would like to thank Abhijit Bagri whose past experience on TeNs helped in a long way.

References

[1] N. S. 2. http://www.isi.edu/nsnam/ns/.

[2] K. G. Ashish Raniwala and T. cker Chiueh. *Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks*. ACM SIGMOBILE Mobile Computing and Communications Review, v.8 n.2, April 2004.

[3] Nitin H. Vaidya. Chandrakanth Chereddi, Pradeep Kyasanur. *Design and implementation of a multi-channel multi-interface network*. second international workshop on Multi-hop ad hoc networks: from theory to reality, May 2006.

[4] Opnet IT Guru Academic Edition. *OPNET Technologies, Inc.* http://www.opnet.com/services/university/itguru_academic_edition.html.

[5] P. Kyasanur and N. H. Vaidya. *Capacity of Multi-Channel Wireless Networks: Impact of Number of Channels and Interfaces*. ACM Mobicom, 2005.

[6] P. Kyasanur and N. H. Vaidya. *Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks*. IEEE WCNC, 2005.

[7] P. Kyasanur and N. H. Vaidya. *Routing and Link-layer Protocols for Multi-Channel Multi-Interface Ad Hoc Wireless Networks*. SIGMOBILE Mobile Computing and Communications Review, Volume 10, Number 1, pages 31-43, January 2006.

[8] C. Chereddi. Pradeep Kyasanur, Jungmin So and N. H. Vaidya. *Multi-Channel Mesh Networks: Challenges and Protocols*. IEEE Wireless Communications, April 2006.

[9] J. Padhye. R. Draves and B. Zill. *Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks*. ACM Mobicom, 2004.

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

[11] The Enhanced Network Simulator. *Indian Institute of Technology Kanpur*. http://www.cse.iitk.ac.in/users/braman/tens/.