An Improved Hop-Count Metric for Infrastructural Wireless Mesh Network

Paramjeet Kaur Bedi[1], Mahendra Singh Aswal[2], Praveen Kumar[3]

Paramjeet Kaur Bedi (Corresponding Author)[1]
Computer Science
Singhania University
Rajasthan, India

Dr. Mahendra Singh Aswal[2]
Computer Science
Gurukul Kangri University
Haridwar,India

Dr. Praveen Kumar[3]
Computer Science
GB Technical University,
U.P.,India.

Abstract - Wireless mesh networks (WMNs) consist of mesh routers and mesh clients, where mesh routers have minimal mobility and form the backbone of WMNs. They provide network access for both mesh and conventional clients. In a WMN, it is possible to reach an IGW or any mesh router (MR) through multiple paths. However this may lead to an undesirable situation in which the best paths may degrade due to the load, consequently resulting in suboptimal performance. To this end, we propose a reactive Hop-Count based multi-path routing protocol which is aware of congestion and location of its immediate neighbors for an Infrastructural Wireless Mesh Network. With support of congestion-aware mechanism, nodes can choose links which have enough available bandwidth for incoming flow's requirement. The main consideration is on the study of routing metric with fast route discovery, minimum hops, minimum delays, maximum data rates and minimum error rates. Simulation results, show that our proposed routing metric is better than existing ETX and ETT metric used in routing.

Keywords - wireless, mesh, multipath, hop-count, routing, metric, congestion-aware.

I. INTRODUCTION

As various wireless networks evolve into the next generation to provide better services, a key technology, wireless mesh networks (WMNs)[1], has emerged recently. 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). This feature brings many advantages to WMNs such as low up-front cost, easy network maintenance, robustness, and reliable service coverage.

With the envisaged application of WMN[2] in providing broadband wireless internet access to end users, the network should have high capacity and enough bandwidth to support the requirements. Any routing algorithm designed for a WMN will aim to invariably choose the best links while finding routes from any router to IGW. Consequently, certain nodes or links can be heavily loaded while some nodes/links are seldom used. Even a routing protocol that periodically monitors links/paths cannot help in such a case, because it may result in too many oscillations and lead to an unstable operation of the network. Moreover, in a WMN, the traffic that is heterogeneous in nature and comprising of significant multimedia content, supporting the desired Quality of Service (QoS) becomes an important requirement. Thus, recalling the chief functionality of a WMN as the means of extending internet connectivity, any proposed solution must efficiently balance the traffic in the network and yet meet the application demands.

Researchers have proposed many metrics for WMN and applied them in QoS-aware routing[4][5][6][7]. However, QoS-aware routing sometimes introduces a sophisticated and unstable algorithm, a lot of overhead and much complexity for measuring the link QoS metrics as well as for calculating, establishing and tearing down the paths. As far as our knowledge, Hop Count metric[8] still predominates in routing implementations these days due to its simplicity and stability. The main drawback of Hop Count metric is that it doesn't allow choosing highest throughput paths. In this paper, we design a mechanism that can detect the congestion risk over links of each node to support Hop Count metric to overcome its disadvantage.

Our basic idea for this strategy is to deploy the WMNs with certain kind of geometric graph as the network topology, and then design a routing protocol by exploiting this graph's routing properties. Our proposed routing protocols is inspired by existing studies on Hop-Count based routing and bandwidth estimation techniques. In our study, we design a mechanism to permit Mesh Routers to evaluate their consumed bandwidth to support Hop-Count based routing while being aware of congestion risk. Our proposal therefore can improve the quality of service of real-time application as well as fulfilling traffic engineering aspect. Our proposal also permits to change how packet is forwarded, without affecting the routing algorithm, in other words, packets will change their routes without invoking routing algorithm to tearing down the old path and establishing the new one so it can save time and overhead. We also propose a route splitting algorithm to provide load balancing in WMN. In simulation part, we show that our proposal can improve network performance in a sense that it can avoid congestion, provide better performance than ETT[9][10] and ETX.

The rest of this paper is organized as follows: Section II lists related works and the inspiration leading to our idea. Section III presents network model and assumptions. Section IV depicts the proposed design and algorithm. Section V introduces congestion-aware and load balancing mechanism that are applied at each node to fulfill traffic engineering. Section VI shows the performance analysis of our proposed scheme using NS2. Section VII concludes main points of this paper.

II. RELATED WORK

Traditional routing protocols designed for MANETs may actually create many hotspots, and lead to subdued performance. For example popular routing protocols for Ad Hoc Networks such as AODV[11], DSR[12] decide the routes solely based on the number of hops. Recent work [13] [14] has revealed that shorter hop routes prevent the network from realizing higher performance when the wireless interface has multi-rate capability. They present a source routing protocol (LQSR) [15] for multi-radio multi-channel WMNs using a novel routing metric, Weighted Cumulative Expected Transmission Time (WCETT). This metric enables the nodes to choose the best possible routes that have good mix of channel variant hops and high bandwidth links. However they do not consider load balancing i.e. the traffic concentration and congestion on certain paths. Another problem is if a node has good link to the next hop but it is heavily loaded then there is no use in selecting a path to IGW through such a node. Thus their protocol may not be scalable to the size of a community WMN .

Yang et al. [16] showed that WCETT can in fact create routing loops in certain situations. They further propose another routing protocol (LIBRA) considering the intra and interflow interference. Even this protocol does not focus on traffic congestion buildup along good paths. Ramachandran et al. [17] propose spanning tree based protocol AODV-ST that modifies the AODV protocol to include Expected Transmission Time (ETT) as the routing metric. Each AP routes traffic through a primary gateway that has least end-to-end delay. In order to estimate the least loaded gateway, periodic RTT probing is performed which is expensive.

Comparing to the existing studies, our focus in this paper is to design a routing metric such that, it reduces transmission delay by choosing the best path in terms of distance and load. So, it can be considered as a fair-routing protocol that distributes the traffic evenly.

III. NETWORK MODEL AND ASSUMPTIONS

Our proposal is based on AOMDV designed primarily for highly dynamic ad hoc networks. However, the main design of AOMDV has been modified to adapt to the stationary nature of nodes in a Wireless Mesh Network. In our research, we propose to explore the shortest path and load balanced multipath routing in WMN. A performance study of the proposed routing algorithm has been carried out in order to compare with the traditional algorithms.

Following are the assumptions taken:

A. Assumptions

1. As in our study, we consider an Infrastructural WMN. We consider that the area is covered by a set of equal-sized hexagonal cells, which is similar to the typical scenario in cellular networks. Each node(mesh router), is placed at each vertex point. One node is placed at the center, which is connected to all six nodes. We utilize the resulting graph for forwarding the packets. We use the location information such as the

coordinates to forward the packets. The location of the source, destination and neighboring nodes, along with the load and link status of all its six neighbor helps in making the forwarding decisions.

2. We use the properties of a regular hexagon i.e. The sides and interior angles of a regular polygon are all equal and the perpendiculars drawn from the center of a regular polygon to each of its sides are all equal.
3. Using these properties we can find the x,y coordinates of all its vertices. E.g., if we consider that the center of a regular hexagon is 0,0 i.e. origin and the radius to any of its vertex is 5, then its x coordinates are determined by rcos60° and y coordinates by rsin60° so, all vertices are known, as shown in the figure below.
4. Whole Network is divided into clusters. The Gateway is placed at the center of each cluster. Default Gateway DG, indicates that the node and the Gateway belongs to the same cluster.

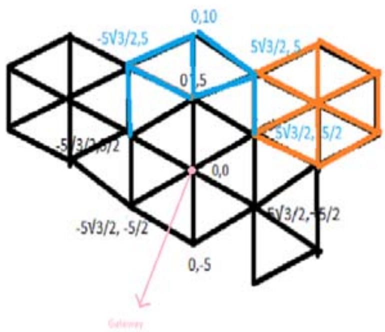

Figure 1: Hexagon showing coordinates construction.

5. We assume that each node has knowledge of all its six neighbors about their link and load status. Each node periodically sends beacon messages to all its neighbors to get their link status. The link status of all the six neighbors is recorded by each node. They also have a record of how much hop away, they are from the gateway, as each hop is of same distance. In order to minimize the collision probability, each node should adjust its power to a level that is able to reach its six neighbors and no more.
6. We also assume that the antenna's we are using at router level can work in two modes: omni-directional and directional[18]. If the node has nothing to transmit, then to detect any incoming signals, the antenna works in omni-direction and when the node need to transmit something to the other node, directional antennas are used so that data could be sent to only the one which is in desired direction with better signal quality.

IV. PROPOSED DESIGN AND ALGORITHM

Our algorithm uses nodes' locations as their addresses, and forwards packets (when possible) in a greedy manner towards the destination. To uniquely identify each address, the coordinates of each along with gateway's IPAddress is sent. These virtual coordinates need not be accurate representations of the underlying geography but, in order to serve as a basis of routing, they must reflect the underlying connectivity. Thus, we construct these virtual coordinates using only local connectivity information. Since local connectivity information is always available (nodes always know their neighbors), this technique can be applied in most settings. Destinations in packets are virtual coordinates.

Our protocol considers the location of the destination; the node which is in x-y direction of destination and is lightly loaded is used. If that node is not lightly loaded then the node next nearer to it with lighter node is picked for forwarding. The path once decided can be changed next time if some of the intermediate node fails or is heavily loaded.

A. Route Discovery

The Route Discovery, begins by determining the location of the destination. A packet is sent to the node which is in same direction as is the destination and is lightly loaded. The path once constructed is cached for further communication. Before sending any packet in the learnt path, each node checks the link status and load of next node in the path. If link to a node in the learnt path fails or is heavily loaded then, the intermediate node sends the packet to the surviving node through the same process, i.e. checking the load and direction of the node. If the destination is a Gateway, then the node which has the minimum of both hop count and load is selected. If the whole network is heavily loaded, then only the location information is used for packet transmission and beacon packets are stopped. Only early link failure notification is used by taking feedback from physical layer of OSI model.

Consider gateway as the point of origin and each node is given x-y coordinate on the basis of their direction from the origin. The objective of this algorithm is to use shortest path, load balanced and fault tolerant delivery of packets with minimum possibility of collisions. Our algorithm is as follows:

ALGORITHM 1

Step1. Check the destination and determine its coordinates.
Step2. if next hop is the destination, complete transmission with it.
Step3. if destination is a Gateway, then
(a)Repeat Step (i) and (ii) until destination is reached.
(i)choose the neighbor with minimum hop- count and lighter load.
(ii) transmit the packet.

Step4. else check the destination ipaddress of the gateway,
(a) if the destination belongs to the default gateway (DG) then,
(i)Repeat Step (ii) and (iii) until destination is reached
(ii)compare destination's coordinates and current node's coordinates and determine the neighbor's load
(iii)select the one which is closest to the destination and is lightly loaded. If the closest node is not lightly loaded, then pick the next closest node for the next hop and transmit;
(b) else
(i) send the packet to its default gateway (DG) choosing the node with minimum hop count and lighter load.
(ii)Gateway either sends the packet directly to the destination gateway (if it exists in the cache), or searches for the destined gateway using traditional AOMDV protocol, and transmit the packets to it.
(iii)after receiving the packets, the destination gateway forward the packet by selecting the one which is closest to the destination and is lightly loaded.
(iv)Goto Step 4(a)(i).

In Step 1, we simply check the destination coordinates. In Step 2, we enquire, if we are next to destination, then simply transmit the packet. Then in Step 3, we see whether the destination is a Gateway, as for gateway, we do not see the coordinates but hop count. As each node also records its hop distance to the gateway and each hop is equidistant (because of the properties of a regular hexagon), so the node with minimum hop count and lighter load is selected for transmission. In Step 4, we compare the destination's gateway ipaddress with the default gateway (DG) ipaddress, if both are same then we compare the destination coordinates with current node's coordinates which is initially the source node, to see in which direction the packet should be forwarded. The node which is closest to the destination and is lightly loaded is selected. If the destination belongs to a different gateway then the packet is sent to the default gateway(DG), from where it is sent to the destination gateway. The destination gateway again upon receiving the packet, forward it to the node which is closest to the destination using the coordinates as described above and is lightly loaded.

This algorithm also takes care of congestion control and load sharing so that nodes nearer to the Gateway are not heavily loaded.

B. Congestion Aware Routing

We introduced a congestion level threshold "µ" to determine whether a particular node in the network is congested or not. The "µ" is the average queue length at a node in a particular path. A mesh router is said to be congested if its congestion level is greater than the threshold, i.e. $\frac{QLi}{tr} \geq \mu$, where *QLi* is the average queue length at a node in a particular path and *tr* is transmission rate at a node.

We use a threshold "µ" to reduce periodic broadcasting of congestion information to the neighbors since if a node is not congested, communication with neighbors periodically results in the wasting the network bandwidth.

The proposed algorithm let the node determine its congestion level itself reducing communication overhead. Each node keeps the track of congestion of its six neighbors with the help of six flags, i.e. $flag\_congestion_i$ for each neighbor. Initially, all flags are set to "0" i.e. not congested. Thereafter, each node 'i' will compute its own $\frac{QLi}{tr}$ periodically. If the ratio, $\frac{QLi}{tr} < \mu$, then we assume that the node is lightly loaded.

But if its congestion is greater than the threshold "µ" , the node will send this information to all its six adjacent neighbors. All the neighbors upon receiving such information will set the flag_ $congestion_i$ to '1' for the node i, which means that, this node is marked congested and no further messages will be sent to this nodes till the congested node reaches half of its threshold i.e. "µ".

After the congested node has reached the queue length to transmission rate ratio $\leq \frac{\mu}{2}$, a packet indicating this is again sent to all its six neighbors, which means, now further transmission is possible on this node again. All the neighbors upon receiving this message, marks this in their table by setting flag_ $congestion_i$ to '0' for the node i, indicating that the route is available again for further transmission.

Algorithm 2

At each mesh router i:

Step 1. Compute $\frac{qel}{tr}$ periodically;

Step 2. If ($\frac{qel}{tr} \geq \mu$ at node i)

then multicast this to its six immediate neighbors and start proc Algorithm 3

Else

Load is balanced, Goto Step 1.

Algorithm 3

Step 1. Compute $\frac{qel}{tr}$ periodically;

Step2. If ($\frac{qel}{tr} < \frac{\mu}{2}$) then,

multicast this to its six immediate neighbors and return to Algorithm 2

Else

Goto Step 1

As shown in Algorithm 2 each mesh router has two states: congested and load-balanced. If $\frac{qel}{tr} \geq \mu$ at node i, the mesh router is in congested state. This information is sent to all its six neighbors and the congested node starts Algorithm 3, where it checks its load again. If the mesh router reaches to half of its µ, i.e. back to its normal load, then the mesh router will notify its recovery to all its previous child nodes. Upon receiving any signal for congestion, the mesh router changes the direction of the packets to next best available node in terms of congestion and link status.

V. RESULT ANALYSIS

The proposed simulation in NS2(NS2 2.29) [19][20] is implemented in Linux with some modification to support congestion-aware mechanism described in section 4. We consider the static wireless mesh network with a number of nodes arranged in the area of 1000x1000 m^2. The topology is a hexagonal topology comprising of 36 nodes, in which each node is attached to six other nodes. The 802.11 MAC protocol is used in Data/Ack mode. Additionally, simulation uses a UDP traffic mode with a Constant Bit Rate (CBR) traffic. All MR has a fixed transmission range of 250m.

In our simulation study, we compare Hop-Count Metric with other two metrics naming ETX and ETT. In order to compare with an existing routing metric, such as ETT and ETX, the same traffic model and protocol of their implementation were used. Source nodes are randomly selected among the common nodes at each runs. In this simulation, the multi channel is not considered.

First of all, we study how the traffic is routed from the MR source to IGWs while traffic load gradually increases on a specific path. We consider the MR 'D', 'L' AND 'Y' where each MR originates and forwards the traffic to DG (the Default Gateway). We choose UDP packets with 1024-byte payload to stand for real-time applications. When the traffic load is 160 packets per second from the nodes O, D and Y, it follows the route D-E-R-DG, L-J-DG, Y-S-R-DG as shown in Figure 2(a). These are the shortest path from MR's to DG. Figure 2(b) shows the path of traffic when load of each node is 310 packets per second. At node Y, the traffic is now diverted to Y-S-U-DG and at node D, the traffic is diverted to D-E-F-DG. The reason is that when the rate was 310 packets per second, the traffic over link R-DG has reached the critical threshold "µ". So the MR 'R' forwarded a link_congestion flag to all its neighbors so that no more future traffic is send to this node.

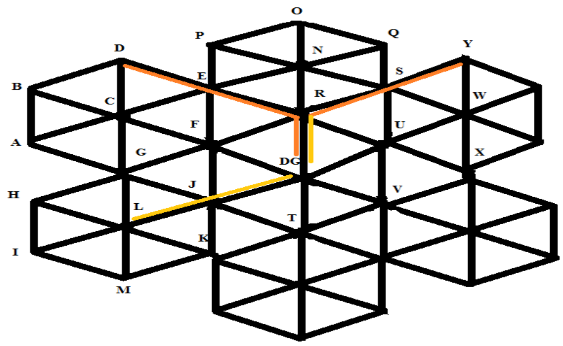

Figure 2(a)Traffic Routed from MR's to Default Gateway with congestion aware mechanism

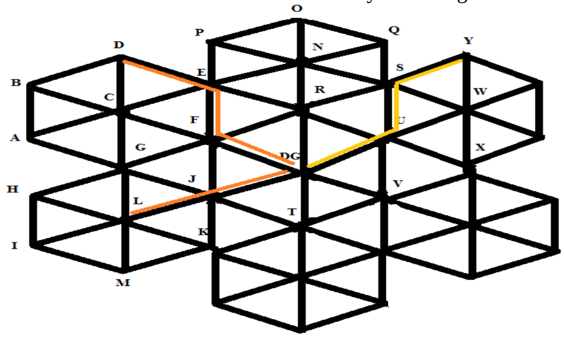

Figure 2(b) Traffic Routed from MR's to Default Gateway after getting congestion notification

Next, we vary the traffic rates to measure throughput of these MRs by increasing number of flows along with time. We can see a big improvement in the throughput of nodes in our proposal (Figure 3) as compared to ETX that does not consider about different transmissions or packet sizes and ETT that does not consider the load balancing of WMNs. In our design, each of examined MR can reach its maximum throughput due to the capability of predicting congestion risk and sharing load among multiple paths to protect links from overloaded.

We study the performance of the network by measuring the sum of throughputs achieved by the individual nodes in the network. We considered 4 simulations each with varied load with 4 pkts/sec to 16 pkts/sec. The simulation results indicates that this routing metric is better than the other routing metric. As it can be seen in the graph, the Hop-Count metric shows that good performance is achieved in the throughput of the whole network. From the graph, we observe that for non-congested scenarios (offered load <8 Mbps), all three schemes are able to sustain the offered load. In other words, our Hop Count metric matches the performance of the other schemes in low congestion environments. With the increase of congestion (offered load >8 Mbps), SampleRate is affected by the congestion-related packet losses and, thus, begins to use lower data rates. But our proposed protocol checks the congestion in its route and switches to an alternate route, which reduces these packet losses and increases the overall throughput.

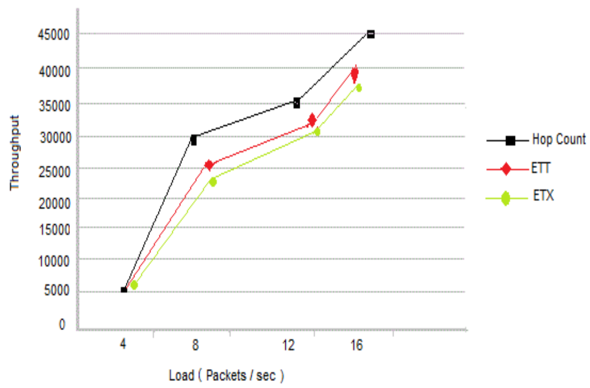

Figure 3 : Throughput analysis

We also measured the End to End Delay by comparing the number of packets sent and number of packets received. For measuring the End-to-End Delay, we considered the same 5 simulations each with varied load with 4 pkts/sec to 16 pkts/sec. By comparing the figures obtained and plotting it on graph (as shown in Fig. 4), we found that the proposed metric has much lower End-to-End Delay than the other routing metrics. We also found that the End-to-End delay is also less when some node reaches its congestion threshold, as our protocol changes its path to the next best path available from the point of congestion. We believe that our congestion measurement technique can be used to design new solutions that perform well under congestion scenarios. However further experiment is needed to confirm and validate the result.

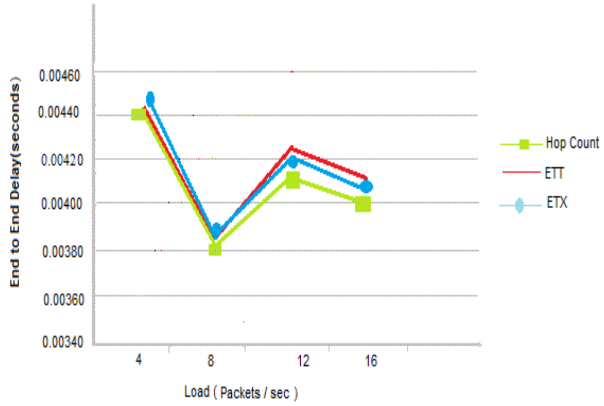

Figure 4: End to End Delay Analysis

VI. CONCLUSION

Routing in any network has a great impact on the overall network performance, thus a routing protocol or algorithm for WMN should be carefully designed taking into account the specific characteristics of the network. We propose a routing strategy, that has the following key contributions: (i) We defined a way to assign coordinates to all the nodes in a network and locate any node using those coordinates. (ii) We defined a way to transmit a packet using shortest path. (iii) We designed a multi-path Hop-Count based routing metric that can be effectively applied for WMN architecture due to its simplicity, stability and practicability. (iv) We design a mechanism that can support our Hop-Count based routing to predict congestion and avoid it.

REFERENCES

[1] "Wireless mesh networks: a survey", Ian F. Akyildiz , Xudong Wang, Weilin Wang (Sept. 2005), Computer Networks 47 (2005) 445–487
[2] "Multipath Routing in Wireless Mesh Networks", Nagesh S. Nandiraju, Deepti S. Nandiraju, and Dharma P. Agrawal, 1-4244-0507-6/06/$20.00 ©2006 IEEE
[3] "Wireless mesh networks: current challenges and future directions of web-in-the-sky," N. Nandiraju, D. Nandiraju, L. Santhanam, B. He, J. F. Wang, D. P. Agrawal, IEEE Wireless Commun., vol. 14, no. 4. pp. 79-89, August 2007.
[4] Koksal, C., Balakrishnan, H " Quality-aware routing metrics for time-varying wireless mesh networks". Selected Areas in Communications, IEEE Journal on 24(11) (Nov. 2006) 1984-1994
[5] "QoS-Aware Routing Based on Bandwidth Estimation for Mobile Adhoc Networks", Lei Chen, Heinzelman W.B, IEEE Journal on Selected Areas in Communications, 2005
[6] "Interference-aware topology control and QoS routing in multi-channel wireless mesh networks," Jian Tang, Guoliang Xue, Weiyi Zhang,Proceeding,MobiHoc '05 Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing.
[7] "Incorporating QoS Awareness in Routing Metrics for Wireless Mesh Networks", Ishita Bhakta, Koushik Majumdar, Ayan Kumar, In the proceedings of World Congress on Engg. 2010 Vol I.
[8] "Hop-Count Based Congestion-Aware Multi-path Routing in
[9] Wireless Mesh Network", Hung Quoc Vo, Choong Seon Hong, IITA-2006-(C1090-0602-0002) Richard Draves, Jitendra Padhye, Brian Zill, "Comparison of Routing Metrics for Static Multi-Jop Wireless Mesh Networks", Proc of SIGCOMM 2004
[10] "Performance Study of Wireless Mesh Networks Routing Metrics",S.Waharte, B.Ishibashi, R.Boutaba, 978-1-4244-1968-5/08/$25.00 ©2008 IEEE.
[11] C. Perkins, E. M. Royer, and S. Das, " Ad-hoc On-demand Distance Vector (AODV) Routing", IETF RFC 3561, 2003
[12] J. Broch, D. Johnson and D. Maltz, "The dynamic source routing protocol for mobile ad hoc networks", IETF Internet Draft, Oct. 1999
[13] Richard Draves, Jitendra Padhye, Brian Zill, "Comparison of Routing Metrics for Static Multi-Jop Wireless Mesh Networks", Proc of SIGCOMM 2004
[14] Douglas S. J. De Couto, "A High-Throughput Path Metric for Multi- Hop Wireless Routing, "Proc. Of ACM MOBICOM, 2003.
[15] "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," Proc of MOBICOM 2004
[16] Y. Yang, J. Wang and R. Kravets, "Designing Routing Metrics for Mesh Networks," IEEE Workshop on Wireless Mesh Networks, WiMesh, 2005
[17] K. Ramachandran *et al.*, "On the Design and Implementation of Infrastructure Mesh Networks," IEEE Workshop on Wireless Mesh Networks (WiMesh), 2005.
[18] " Opportunities and Challenges for Mesh Networks Using Directional Antennas",Guoqing Li, L. Lily Yang, W. Steven Conner, Bahareh Sadegh.
[19] On NS2 information available at " http://www.isi.edu/nsnam/ns/".
[20] The *ns* Manual, formerly *ns* Notes and Documentation