Quality of Service using Adaptive Traffic Balancing in Wireless (Ad Hoc) Mesh Networks

DR. TRILOK CHAND ASERI[1]
ASLAM QURESHI[2]

Department of Computer Science and Engineering
Punjab Engineering College (Deemed University)
Sector-12, Chandigarh-160012 (India)
[1]a_trilok_chand@yahoo.com
[2]aslam_q2001@yahoo.com

Abstract. Wireless Mesh networks exploit multi-hop wireless communications between Access Points to replace wired infrastructure. For improving the performance of Quality of Service in face of unreliable wireless medium, an adaptive traffic balancing using traffic distribution algorithm has been proposed. In this paper, an adaptive traffic balancing is described to change the verification rule of the traffic state at each node dynamically, according to the number of collisions detected by the node. The simulation has been done to evaluate the developed algorithm. The results show that after deploying adaptive traffic balancing, performance is improved and the system resources are used more efficiently.

Keywords: Traffic Balancing, Quality of Service (QoS), Adhoc Networks, Mesh Networks.

(Received July 21, 2008 / Accepted December 15, 2008)

1 Introduction

Several access points may be put into wireless Ad Hoc networks for people connecting to World Wide Web (WWW) or other internet services. These kind of wireless networks are called Wireless Mesh Networks. The Wireless Mesh Network (WMN) is a new broadband Internet access technology. The competition with other broadband technologies, including cable, digital subscriber line, broadband wireless local loop and satellite Internet access, is stiff, but WMNs have significant advantages, making them a viable alternative. Optimal load balancing across a mesh or a network is a known hard problem. Load balancing as an optimization problem has been described in [10]. Efficient load balancing in wireless networks [9] becomes an even more challenging problem due to the limitations on available bandwidth and unreliability of wireless links. We consider load balancing in wireless mesh networks with stationary nodes. These include Wireless Mesh [2], [7] and Community networks [6], such as the Self-Organizing Neighborhood Wireless Mesh Networks [20].

The rest of the paper is organized as follows. An overview of wireless mesh networks is given in Section 2. Section 3 describes the previous work related to the quality of service in wireless mesh networks. The algorithm developed for adaptive traffic balancing and its performance evaluation is illustrated in Section 4. The simulated results are discussed in Section 5. Section 6 concludes the paper.

2 Overview of Wireless Mesh Networks

Wireless Mesh Networks (WMN) are envisioned to support the wired backbone with a wireless backbone for providing internet connectivity to residential area and offices. The primary advantage of these networks is their rapid development and ease of installation. Unlike traditional WiFi networks where each access point is connected to the wired network, WMNs use the paradigm of mutltihop communication. In WMNs, mesh routers (hybrid version of access points) communicate with the external network (for example Internet) by cooperatively forwarding each others' traffic towards the gateway nodes (mesh points) which are directly connected to the wired backbone. In this paper, the terms access points and mesh routers are used interchangeably. Similar to the wired network where intermediate routers form backhaul and route traffic from one network to another, in a WMN, mesh points and mesh routers forward each other's traffic in order to establish connectivity. Note here that mesh routers and mesh points are similar in design with the only exception that a mesh point is connected to the wired network and thus also called as Internet gateway [1][8].

3 Related Work

Backbone Wireless Mesh Networks (BWMNs) are highly adaptable, scalable, reliable and cost effective, which can be deployed easily in areas where the deployment of wired backhaul is difficult or cost-prohibitive. Figure 1 depicts a possible wireless mesh network scenario. In a WMN, several access points (G1 to G4), which have other communication methods of connecting to the Internet, are deployed, and they do not have to cover all the areas of the network. If the node density is at certain level, most nodes can reach the access points in a few hops to access the Internet. Therefore, WMNs have several significant advantages [18]: (1)Very high coverage levels with very low initial investment. (2) Excellent spectral efficiency. (3) Complete flexibility in service delivery. Unlike wireless Ad Hoc networks, traffic in WMNs concentrates in the area around the access points and the throughput of each node decreases as O(1/n), where n is the total number of nodes in the network [11].

However, if there is more than one access point, a mobile node is able to choose a feasible path leading to one of the access points. Here we assume that a protocol takes responsibility for coordinating among the access points. Therefore, a node can change the path that may lead to different access points, without worrying about the interruption of ongoing communications (similar to handover in cellular systems). The access points in WMNs can be added one at a time, as needed. Adding more access points will increase not only the capacity of the network but also its reliability. In most applications, quality of service is an essential component. Quality of Service (QoS) aspects include bandwidth, delay and delivery guarantees. It has been proposed that to achieve reliable QoS, wireless Ad Hoc networks will require traffic engineering capabilities, and providing these capabilities will require the cooperation of three components: a QoS-capable medium access control protocol; a resource reservation scheme; and a QoS routing protocol [14].

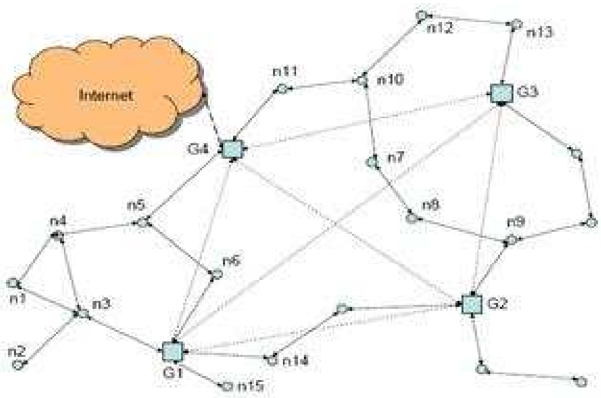

Figure 1: Example of Wireless Mesh Networks

Most early research on QoS focused on designing QoS routing protocols [3] [4] [12]. Later, more researchers found that it is difficult to satisfy QoS guarantees solely in higher layers without support from the Medium Access Control (MAC) layer [19] [16] [17] [5] move further into the physical layer to ensure certain QoS requirements. Research in [13] provides a theoretical analysis for QoS implementation in wireless (Ad Hoc) Mesh networks, if global information can be acquired. IEEE 802 has defined a new MAC protocol (IEEE 802.11e) to support QoS, and more research groups are still working on the MAC layer to address multichannel problems and be more efficient [15]. A lot of research still needs to be done to address QoS implementation, which may combine techniques from all layers.

4 Performance Evaluation of Adaptive Traffic Balancing

4.1 Traffic Load

When more than one access point is put into the WMN, uneven traffic load problem appears time to time. The major reason for this problem is the routing algorithm. In most of routing protocols, control information and routing packets have higher priority than data packets. If on-demand routing protocol is used, the first reply for route request usually goes through the shortest or closer paths. Without other weights to evaluate each hop, the shortest path or closer one is chosen as the default path. In certain period, most end users will be close to one access point and most of the traffic load concentrates on this access point. High packet loss rate and long delay caused by the overloaded access point lead to a poor system performance. For an instance, in Figure 2, due to the traffic pattern and routing protocol most traffic load goes to access point G1 even some of them can be routed to the nearby access point G2.

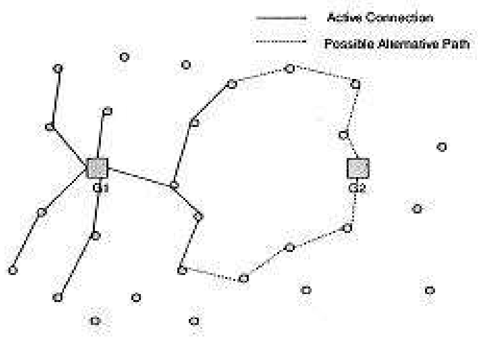

Figure 2: Traffic Load to access points

4.2 Traffic Balancing

Traffic balancing using traffic distribution algorithm (refer Appendix-A) has been used for solving uneven traffic load in wireless mesh networks due to more collisions. When the traffic load is high, the middle of the network (the area around node E) will be congested (more collisions may take place and data packets have to be retransmitted), while other areas (such as the area around the dotted line in Figure 3) remain in a less loaded state. Because of the poor path selection, the overall system utilization is far below the theoretical limit, even if the traffic load becomes very high. Because the queue length of each node is fixed, when the queue is full, new incoming data packets have to be dropped. Moreover, frequent transmission collisions lead to a large number of retransmissions and longer backoff time, which cause packets to wait longer in queues. Some packets are dropped because they exceed their allowed lifetime. In Figure 3, nodes A, C, and F communicate with nodes B, D, and G respectively. According to the reactive routing algorithm, the broadcast route requests and replies have higher priority than data packets. All of them will pass through node E in order to achieve the shortest distance.

Instead of selecting path 1, indicated by the solid line between nodes A and B in Figure 3, path 2 - indicated by the dotted line - is used for packet delivery for node A and B, and gives node E a good chance to fully support communication between nodes C and D, and nodes F and G. This can either expand the system's ability to support more communications under the same performance level, or improve the performance of the system. A few issues need to be addressed before discussing the solution. First of all, each node should have the ability to record the usage of the medium around itself. In fact, all information required can be collected from the MAC layer, because it is monitoring the medium at all times in order to send or receive packets. The measured results will help the node decide if the medium in its area is overloaded or not. Our protocol function is implemented in such a way that each node records the state of the medium in the past n milliseconds. If a node detects that the received power is greater than the noise threshold for a certain period (equal to or longer than the time needed to transmit the smallest frame), the medium is recognized as being used by other nodes. The duration of the state that the medium is occupied is recorded and accumulated in order to calculate the percentage that the medium is busy. In the MAC layer of each node, a linked list is required to record every busy period of the medium.

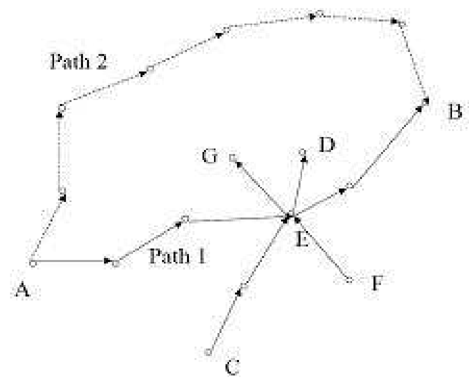

Figure 3: Selected Paths According to the Routing Protocols (Bold line indicates those chosen by Dynamic Source Routing (DSR))

The procedure for medium measurement is as follows:

Each node continues to sense the received power level. Once the power level has gone over a certain threshold (noise threshold), the medium is assumed to be used for a transmission, which could be a transmission by the node itself. The MAC layer functions then record the start time (Tstart_time) and the duration of the transmission (Tduration), and add them to the linked list as a new element. Meanwhile, all the elements in the linked list are checked, and those elements that do not satisfy the following condition (Equation 1) are removed:

$$Tcurrent\_time - (Tstart\_time + Tduration) < n \quad (1)$$

It also means that the medium occupations happened n milliseconds ago are removed from the linked list because they are not counted for calculating the medium usage. When the node forwards a route request, the MAC layer functions check the linked list again and accumulate the elements that obey the previous condition, as follows (Equation 2):

$$Tbusy = \Sigma Tduration \quad (2)$$

The node can then verify the percentage of medium usage in the past n milliseconds time periods by (Equation 3):

$$MeasuredMediumUsage = Tbusy/n \quad (3)$$

Now that it has the value of the measured medium usage, the node can decide if it is in a busy area or not. Overall, the MAC layer protocol updates the linked list once the medium is used, and calculates the medium usage when a route request needs to be forwarded. The choice of the value of n depends on the types of traffic in the network. If the traffic is bursty, the n (measurement period) should be small. Otherwise, the n should be large. The second issue is the introduction of an additional bit or byte to indicate the medium usage in the header of the route request, which is set by the sender.

Thus, two parameters have to be defined before the implementation of Traffic Balancing: the medium usage threshold p and the measurement time period n. The measurement time period has already been mentioned. The medium usage threshold is used to verify whether the medium is busy or not. There are two ways to implement Traffic Balancing, depending on who makes the routing decision: the sender or the intermediate nodes.

a) Intermediate node decision: This solution tries to find a path without any congested intermediate nodes and needs two types of routing requests: low or high priority, requiring one bit in the packet header to distinguish them (0 for a low priority route request and 1 for a high priority route request). For the intermediate nodes: When a node receives a general route request (low priority route request), it first checks the medium usage around it in the past n milliseconds. If the medium usage is higher than the medium usage threshold, the node will ignore this route request. Otherwise, it processes this route request the way it usually does for any reactive routing protocol. When a node receives a high priority route request, it will process it regardless of the medium usage around it. For the sender: To initiate a path, the node first broadcasts a low priority route request. If there is no response after a back-off time (T), a high priority route request is generated and broadcasted.

b) Sender decision: This solution tries to find a path with the smallest number of congested intermediate nodes, and needs one byte in the route request packet header (overThresholdCounter) to indicate the number of nodes in the path that are overloaded (assuming a maximum number of nodes on any given path). The overThresholdCounter is set to 0 by the sender before the route request is sent out. For the intermediate nodes: When a node receives a route request, it calculates the medium usage around it. If the medium usage is over the medium usage threshold, the node increases the overThresholdCounter by one. Otherwise, it simply follows the regular procedure and forwards it. For the receiver: After having received the route request, the receiver feeds back the threshold counter to the sender via the route reply.

For the sender: The sender initiates the route request with an overThresholdCounter equal to 0. Upon receiving the route replies, the sender chooses the route with the smallest overThresholdCounter. If more than one path has the same smallest overThresholdCounter, the one with fewer hops is chosen. If more than one path has the same overThresholdCounter and the same number of hops, the sender chooses the one that arrives first.

4.3 Adaptive Traffic Balancing

If traffic balancing is able to adjust the medium usage threshold according to node mobility or congestion state, the performance can be largely improved. Adaptive Traffic Balancing to adjust the medium usage threshold dynamically, according to the number of collisions seen by the node. As a result, traffic load can be distributed more evenly and system resources can be utilized more efficiently. For the purpose of changing the medium usage threshold dynamically, according to node mobility, certain parameters are required to designate node mobility or usage condition of the medium. During the simulations, it was noticed that the number of collisions that can be seen by a given node over a certain period can be applied to specify the node mobility or medium usage conditions more precisely than other parameters. Although a node cannot perceive all collisions that happen in its interference range, the frequency of observed collisions is still a good parameter for indicating the mobility around it. Therefore, Adaptive Traffic Balancing measures the medium usage and number of collisions over a certain period. If the medium usage exceeds the threshold, the overThresholdCounter is increased. Otherwise, the route request is simply forwarded to the next node.

Adaptive Traffic Balancing records the number of collisions in the past n milliseconds in a similar way that the medium usage at the MAC layer is recorded. In this case, a linked list is required. When a collision or a capture is detected, an element is added to the linked list to note the time (A capture happens when over two transmissions starts simultaneously and one transmission can be received correctly while others are not). Meanwhile, the linked list is updated to remove those elements that satisfy the following condition (Equation 4):

$$Tcurrent\_time - (Tstart\_time > n \qquad (4)$$

The collisions or captures, which is over n milliseconds old, are removed from the linked list because they are not counted for deciding the medium usage threshold. When a route request needs to be forwarded, the linked list is checked again, and the number of elements that obey the previous condition is calculated as well. If the number of collisions in the past n milliseconds is known, a suitable medium usage threshold is then selected.

For intermediate nodes: Each time a route request is received by a relay node, the node checks the number of collisions in the past n milliseconds. If the number is less than 10, the medium usage threshold is then set to 0.9, assuming that the node mobility in the network is low (static). If this number is between 10 and 50, the medium usage threshold is set to 0.7. That assumes that the node mobility is moderate (node mobility is less than 5m/s). Otherwise, the medium usage threshold is set to 0.5 (assuming node mobility is high, with a maximum moving speed of 20m/s). After the medium usage threshold has been set, the node compares the measured medium usage with the threshold and decides whether or not to increase the overThresholdCounter.

5 Results and Discussion

In this research, the solutions are simulated in Network Simulator (NS2). Adaptive Traffic Balancing shows fairly constant improvements in terms of both packet delivery rate and average delay, even compared to other traffic distribution solutions.

Table 1 compares the packet loss among Dynamic Source Routing (DSR), Traffic Balancing and Adaptive Traffic Balancing for the scenario (maximum speed: 5m/s; network size: 1000x1000 m^2). The total number of packets dropped decreased from 21.71% to 5.85% between DSR and Traffic balancing and the total number of packets dropped decreased from 5.85% to 4.73% between Traffic balancing and Adaptive. Adaptive Traffic balancing does well in distributing the traffic load evenly into the network. The number of packets dropped by a router was reduced only minimally.

Table 1: Packet Loss of DSR, Traffic Balancing and Adaptive Traffic Balancing

Protocol	DSR	Traffic Balancing	Adaptive Traffic Balancing
Total number of packets generated	63155	63155	63155
Total number of packets dropped	13708	3693	2989

In Figures 4, 5, 6, 7, 8 and 9, the results show that an Adaptive Traffic Balancing improves system performance further in forms of both packet loss rate and average delay. The improvement with a maximum moving speed of 20m/s is up to 70%-80% for Adaptive Traffic Balancing in the packet loss rate, compared to 50% for Traffic Balancing. Due to the number of collisions seen by the nodes, the medium usage thresholds vary at different nodes. With this additional information at each node, Adaptive Traffic Balancing is able to obtain more information during route discovery and recovery. The chosen path is then more feasible in terms of bandwidth availability. When the maximum moving speed of the nodes is 5m/s, 0.7 is almost the best value for the medium usage threshold, on average; thus the improvement is lower compared to the case when the maximum moving speed is 20m/s. When the node mobility decreases, the improvement due to Adaptive Traffic Balancing is also decreased, because Adaptive Traffic Balancing and Traffic Balancing consume more bandwidth for control information to ascertain a better path. In addition, in the 0m/s scenario, there is almost no collision at all. Adaptive Traffic Balancing stays at one level (here it is 0.9). The improvement in the 0m/s scenario is caused by Traffic Balancing using 0.7 as the medium usage threshold. The improvement due to Adaptive Traffic Balancing is calculated by the following formula (Equation 5)):

$$Improvement = (PLR_{DSR} - PLR_{tb})/PLR_{DSR} \qquad (5)$$

where PLR is Packet Loss Rate, tb is traffic balancing

When the packet loss rate of DSR is about 20%-40%, Adaptive Traffic Balancing gains around 50%, i.e.

the packet loss rate of Traffic Balancing is around 10%-20%.

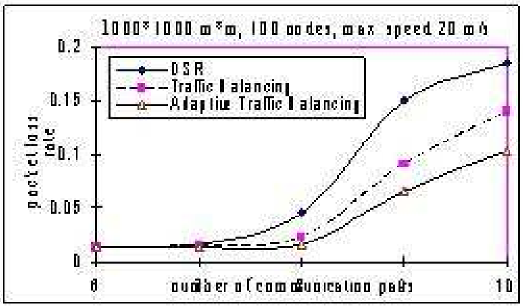

Figure 4: System Performance of Adaptive Traffic Balancing at a Max. Moving Speed of 20m/s [Packet loss rate]

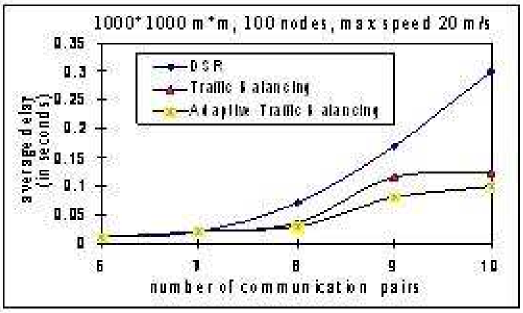

Figure 5: System Performance of Adaptive Traffic Balancing at a Max. Moving Speed of 20m/s [Average delay (in Seconds)]

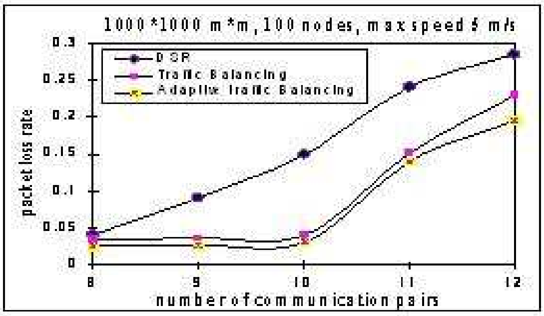

Figure 6: System Performance of Adaptive Traffic Balancing at a Max. Moving Speed of 5m/s [Packet loss rate]

6 Conclusion

Adaptive Traffic Balancing has been proposed to improve system performance in wireless (Ad Hoc) Mesh networks. With knowledge of the load on the medium

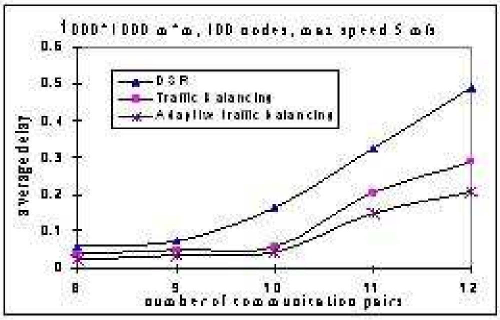

Figure 7: System Performance of Adaptive Traffic Balancing at a Max. Moving Speed of 5m/s [Average delay (in Seconds)]

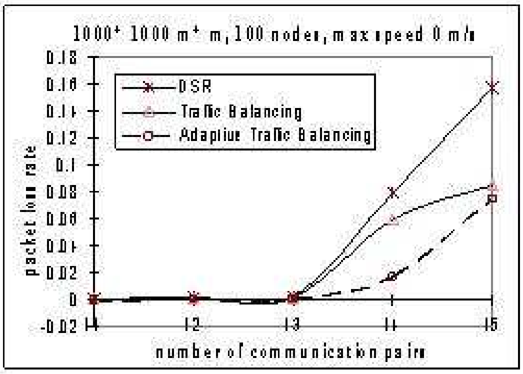

Figure 8: System Performance of Adaptive Traffic Balancing at a Max. Moving Speed of 0m/s [Packet loss rate]

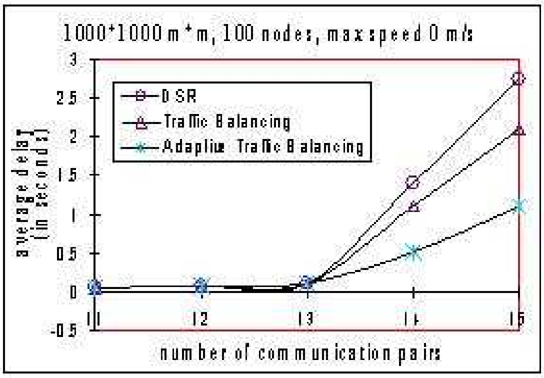

Figure 9: System Performance of Adaptive Traffic Balancing at a Max. Moving Speed of 0m/s [Average delay (in Seconds)]

along all paths, Traffic Balancing exploits unused network resources and routes packets through the appropriate paths. In the proposed Adaptive Traffic Balancing, nodes are able to change dynamically the medium usage threshold intelligently by checking the collision rate in the area, so that the sender collects more accurate network information and chooses a better path. The simulation results show that both the data packet loss rate and the average end-to-end delay can be decreased by over 50% during congestion. With the benefits brought by Traffic Balancing, more connections could be supported with no deterioration in quality of service. Adaptive Traffic Balancing provides a way to force part of traffic load from busy area to light-loaded access points and improve the network efficiency.

References

[1] Akyildiz, I. F., Wang, X., and Wang, W. Wireless mesh networks: A survey. *Computer Networks (Elsevier)*, 47(4):445–487, March 2005.

[2] Bruno., R., Conti, M., and Gregori, E. Mesh networks: Commodity multihop ad hoc networks. *IEEE Communications Magazine*, 43(3), March 2005.

[3] Chakrabarti, S. and Mishra, A. Qos issues in ad hoc wireless networking. *IEEE Communications Magazine*, 39(2):142–148, February 2001.

[4] Chen, S. and Nahrstedt, K. Distributed quality-of-service routing in ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1488–1505, August 1999.

[5] Comaniciu, C. and Poor, H. V. Qos provisioning for wireless ad hoc data networks. In *42nd IEEE Conference on Decision and Control, Maui, Hawaii, USA*, volume 1, pages 92–97, December 2003.

[6] Couto, D. S. J. D., Aguayo, D., Bicket, J., and Morris, R. A high-throughput path metric for multi-hop wireless routing. In *Ninth Association for Computing Machinery (ACM) Annual International Conference on Mobile Computing and Networking (MobiCom'03), San Diego, CA, USA*, pages 227–240, Sept 14-19 2003.

[7] Draves, R., Padhye, J., and Zill, B. Routing in multi-radio, multi-hop wireless mesh networks. In *International Conference on Mobile Computing and Networking (MOBICOM 2004), Philadelphia, PA, USA*, pages 114 – 128, Sept. 26-Oct. 1 2004.

[8] Eastlake, D. E., Sakoda, K., and Rayment, S. Mesh networking task group report (ieee 802.11s), the draft for mesh networking. Technical report, IEEE, USA, November 2008.

[9] Hassanein, H. S. and Zhou, A. Routing with load balancing in wireless ad hoc networks. In *ACM (Association for Computing Machinery) Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 89–96, July 2001.

[10] Jones, E. P. C., Karsten, M., and Ward, P. A. S. Multipath load balancing in multihop wireless networks. In *IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WIMOB 2005)*, pages 158–167, August 2005.

[11] Jun, J. and Sichitiu, M. The nominal capacity of wireless mesh networks. *IEEE Wireless Communications Magazine, Special Issue on: Merging IP and Wireless Networks*, pages 8–14, October 2003.

[12] Lin, C. R. and Liu, J. S. Qos routing in ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1426–1438, August 1999.

[13] Pandya, A. and Pottie, G. Qos in ad hoc networks. In *58th IEEE Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003*, volume 5, pages 3089 – 3093, 6-9 October 2003.

[14] Perkins, D. and Hughes, H. A survey on quality of service support in wireless ad hoc networks. *Journal of Wireless Communication and Mobile Computing (WCMC), Special Issue on Mobile Ad Hoc Networking: Research, Trends, and Application*, 2(5):503–513, September 2002.

[15] Realp, M. and P´erez-Neira, A. I. Decentralized multiaccess mac protocol for ad-hoc networks. In *14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 03), Beijing,China*, volume 2, pages 1634–1638, September 2003.

[16] Sobrinho, J. and Krishnakumar, A. Quality-of-service in ad hoc carrier sense multiple access wireless network. *IEEE Journal on Selected Areas in Communication*, 17(8):1353–1368, August 1999.

[17] Stine, J. and de Veciana, G. A paradigm for quality-of-service in wireless ad hoc networks using synchronous signaling and node states. *IEEE Journal on Selected Areas in Communications*, 22(7):1301–1321, September 2004.

[18] Whitehead, P. Mesh networks; a new architecture for broadband wireless access systems. In *IEEE Conference on Radio and Wireless (RAWCON)*, pages 43 – 46, 10-13 September 2000.

[19] Xiao, Y. and Li, H. Local data control and admission control for qos support in wireless ad hoc networks. *IEEE Transactions on Vehicular Technology*, 53(5):1558 – 1572, September 2004.

[20] Yang, Y., Wang, J., and Kravets, R. Designing routing metrics for mesh networks. In *First IEEE Workshop on Wireless Mesh Networks (WiMesh),*, September 2005.

Appendix - A
Traffic Distribution Algorithm
Notations

- N : Set of network nodes.
- t_{ix} : The traffic sent on link i by node x.
- T_x : Total traffic to be sent by node x in kbps.
- r_x : Number of routes from node x to its connected gateways, i.e., the number of gateways node x is connected to.
- P_n : Priority of node n. Equivalent to traffic routed by n for other nodes in kbps. For example, if node n routes 30 kbps traffic for another node, its priority becomes 30.
- RT_n : Remaining traffic to be sent by node n.
- C_j : Cost of link j. Equivalent to traffic routed on link j in kbps. For example, if link j is reserved for routing 30 kbps traffic, its cost becomes 30.
- D : Max(r_x) for x = 1 to |N|. This is the maximum number of gateways that any node in the network has paths to, from the set of all nodes. This fixes the number of iterations of the algorithm.
- Z : the array containing RT_n + P_n values for all nodes n belonging to N.
- S_x : The current shortest path to a gateway node from node x.

N ← all nodes
L ← links
S ← shortest paths

procedure TrafficDsitribute (N, L, S) {
t_{ix} = [T_x / r_x]
for each n ϵ N {
P_n = 0; RT_n = T_n }
for each l εL {
Cl =1; }
done_count = 0;
Threshold
for iterations = 1 to D {
Sort(Z); //sort array Z in decreasing order
new_count = |N|- done_count
for x = 0 to new_count {
Choose node n_x corresponding to Z[x];
check = 0;
while(check==0){
Compute S_x //current shortest path for n_x
For each i, s.t. li ϵS_x
if(c_i + t_{ix}) $\geq c_i$ {
check = 0;
Remove present path from list of k shortest paths for node n_x }
else {
check = 1;
}
} //end of (while == 0) loop
S_x = n_x; //Assign n_x its shortest path
for each i, s.t. li ϵ S_x{
c_i = c_i + t_{ix};
RT_x = T_x - t_{ix};
if (RT_x = 0)
done_count = done_count + 1;
for each k, s.t. n_k ϵ S_x //nodes lying on the path S_x
P_k = P_k + t_{ix};
for each n_q, s.t. q is from x to (|N| - done_count)
Recompute S_x;
}}}
After computing shortest path find the collision at the intermediate node in this path then Set the threshold value according the collision.

if (medium_usage > threshold_value)
overloaded then choose another path
else
forwarding route request on this path
}