University of New Haven

University of New Haven
Digital Commons @ New Haven

Electrical & Computer Engineering and Computer Science Faculty Publications

Electrical & Computer Engineering and Computer Science

12-2013

Efficient Routing for Wireless Mesh Networks Using a Backup Path

Amir Esmailpour
University of New Haven, AEsmailpour@newhaven.edu

Nidal Nasser
Alfaisal University

Jehad Al Dallal
Kuwait University

Follow this and additional works at: http://digitalcommons.newhaven.edu/electricalcomputerengineering-facpubs

Part of the Electrical and Computer Engineering Commons, and the Information Security Commons

Publisher Citation

Esmailpour, Amir, Nasser, Nidal, & Al Dallal, Jehad (2013) . Efficient routing for wireless mesh networks using a backup path. International Journal of Communication Systems 12/2013; 26(12), pages 1520–1542. DOI: 10.1002/dac.2324

Comments

This is the accepted version of the following article: Esmailpour, A., Nasser, N., & Al Dallal, J. (2013) . Efficient routing for wireless mesh networks using a backup path. *International Journal of Communication Systems* 12/2013; 26(12), pages 1520–1542, which has been published in final form here. Article first published online: 2 MAY 2012

EFFICIENT ROUTING FOR WIRELESS MESH NETWORKS USING A BACKUP PATH

Amir Esmailpour
Department of Electrical & Computer Engineering, University of New Haven
West Haven, Connecticut, USA 06516
aesmailpour@newhaven.edu

Nidal Nasser
Department of Computing & Information Science, University of Guelph, Guelph, Ontario, Canada N1G 2W1
nnasser@uoguelph.ca

Jehad Al Dallal
Department of Information Science Kuwait University
j.aldallal@ku.edu.kw

Abstract

Wireless Mesh Network (WMN) has a proven record in providing viable solutions for some of the fundamental issues in wireless networks such as capacity and range limitations. WMN infrastructure includes clusters of Mobile Ad-Hoc Networks (MANETs) connected through a fixed backbone of mesh routers. The mesh network can be constrained severely due to various reasons, which could result in performance degradation such as a drop in throughput or long delays. Solutions to this problem often focus on multipath or multichannel extensions to the existing ad-hoc routing protocols. In this paper, we propose a novel solution by introducing an alternative path to the mesh backbone that traverses the MANET part of the WMN. The new routing solution allows the Mobile Nodes (MNs) to establish direct communication among peers without going through the backbone. The proposed alternative ad-hoc path is used only when the mesh backbone is severely constrained. We also propose, for the first time in WMNs, using MNs with two interfaces, one used in the mesh backbone communication and the other engaged in the ad-hoc network. A scheme is presented for making the MN aware of link quality measures by providing throughput values to the AODV protocol. We use piggybacking on route reply messages in AODV to avoid incurring additional costs. We implemented our solution in an OPNET simulator and evaluated its performance under a variety of conditions. Simulation results show that the alternative ad-hoc path provides higher throughput and lower delays. Delay analysis show that the throughput improvement does not impose additional costs.

***Keywords**: Wireless mesh network, mobile ad-hoc network, backbone path, ad-hoc path.*

1. Introduction

Wireless Mesh Network (WMN) has become the hype of wireless deployment in urban and rural areas with poor infrastructure. WMN is comprised of two types of equipment: Wireless Mesh Routers (WMRs) and Mobile Nodes (MNs). WMRs are deployed at fixed locations and connected through wireless links to form the backbone of WMN [1]. MNs form clusters of ad-hoc networks that connect to the mesh backbone through access links. Access Mesh Routers (AMRs) are a subset of WMRs that connect to MNs directly on the access side. AMRs in the backbone have two wireless links, one to connect to other WMRs and the other to connect to user devices. Routing in WMNs is challenging due to unpredictable behavior of wireless links caused by interference, noise, fading, and channel propagation. Routing proposals for WMN often focus on ad-hoc routing with some extensions, such as introducing new metrics to reflect wireless link conditions, e.g. Expected Transmission counts (ETX) [2], Expected Transmission Time (ETT), and Weighted Cumulative ETT (WCETT) [3]. These solutions are all concentrated around the mesh backbone network. However, WMN has a major component on the access network with clusters of mobile ad-hoc nodes. In this research study, we focus on the access network rather than the backbone. We propose an efficient routing system for WMN that utilizes both backbone and access links, while introducing a backup path to be used when the backbone is constrained. The WMN backbone is formed by fixed wireless routers; thus, its topology does not change frequently. This allows us to employ link-state routing in the backbone. MNs connected to the WMRs in the backbone could also make direct connection with their peers through their access links to form the Mobile Ad-hoc Network (MANET). MANET topology undergoes frequent changes due to the mobility of MNs. Therefore, an on-demand routing protocol such as Ad-hoc On-demand Distance Vector (AODV) is more suitable for this part of the network. Our proposed routing system provides a solution for routing of WMNs with at least two alternative paths: one through the MANET called the *ad- hoc path* (*ah_path*), and the other through the backbone, called the *backbone path* (*bb_path*). WMRs in the backbone are fixed; therefore, the *bb_path* is more stable than the *ah_path* and has no power constraints. In contrast, the *ah_path* is relatively unstable with power constraints due to the mobility and limited power source of user devices. Hence, the *ah_path* should be used as a backup path only when the primary *bb_path* is not available or is severely constrained. The motivation to use the *ah_path* is clear in at least three situations: first, when the access link contention between the MN and the AMR significantly reduce throughput of the *bb_path*; second, for the handover delay while an MN moves from one cluster to another that could cause a transient outage to the *bb_path*; and third, when the number of hops between the Source MN (S_MN) and the Destination MN (D_MN) are fewer through the *ah_path* than the corresponding *bb_path*. For instance, two MNs in adjacent clusters could communicate directly via the *ah_path* rather than traversing several hops through the corresponding *bb_path*. The proposed routing solution provides the MN with

two paths to choose from: the *bb_path* or the *ah_path*. The MN should have performance information on both paths in order to make a decision regarding which path to choose. This kind of information could be provided to the MN by using link quality metrics in the routing protocol.

The proposed integrated routing system also considers two types of MNs: those with one physical interface and those with two physical interfaces. In the case of MNs with two interfaces, using two different radio frequencies, the MNs will reduce channel contention and improve traffic throughput. MNs will use one interface to connect to the backbone AMR and the other interface to connect to other MNs in the ad-hoc network.

We use AODV as the main routing protocol for the MN and integrate throughput into its source code, in the routing cache and Route Reply (RREP) packet of AODV. We implemented this solution in OPNET modeler 14.5 and show by simulation that AODV performance improves by providing throughput information to regular AODV.

The rest of this paper is organized as follows: in Section 2, we provide related work in the area of routing for WMNs. Section 3 presents the architecture of WMN that is used in this paper. In Section 4, we establish the design principles used in this work. In Section 5, we evaluate the performance of our proposed routing system and show the simulation results. Finally, in section 6, the conclusion and future work will be presented.

2. Related Works

In this section, we present a review of several related papers in the literature in different areas of routing for both MANET and WMNs, as well as some approaches for enhancing WMN routing performance, such as including metrics in the routing protocols for multi-path approaches.

2.1. Wireless and Ad-Hoc Routing Protocols

Traditional routing protocols fall short of meeting the high demand of ad-hoc networks with their unique characteristics. This phenomenon led to the design of new routing protocols exclusively for ad-hoc networks. MANET is characterized by mobility of nodes, limited power supply, and unstable routes. These characteristics result in continuous topology changes that create an enormous amount of overhead, calculations, and flooding by using existing routing protocols. Several new routing protocols have been proposed to improve the traditional protocols when used for ad-hoc networks.

Numerous routing protocols have been proposed for ad-hoc networks in the past few years. Several surveys are available covering and summarizing publications in this area [13, 14, and 15]. Proposals include hierarchical routing, cross-layer designs, clustering, and so on. One of the most common ways to characterize those routing protocols is to divide them into reactive versus proactive groups. Proactive protocols, such as Highly Dynamic Destination-Sequenced Distance Vector (DSDV) [16] and Optimized Link State Routing Protocol (OLSR) [17], keep routes in their routing table and periodically update them. Reactive protocols, such as AODV [18] and Dynamic Source Routing (DSR) [19], on the other hand, work on a need-driven basis, where a route discovery is only initiated based on-demand.

2.2. WMN Routing Protocols

Wireless medium characteristics affect the behaviors of wireless networks such as channel fading, contention, interference, and other physical and MAC layer issues. Therefore, in order to be more efficient, routing protocols for wireless networks should be aware of such lower-layer problems. This point led to the idea of a cross-layer design for routing protocols where the lower-layer characteristics could be communicated to the network layer in the form of new metrics that could be incorporated into layer-3 packet headers. Reviews of cross-layer designs and proposed metrics are presented in [20] and [21]. Iannone [7] introduces new metrics for interference and packet success estimation ratios that are communicated among the physical, MAC, and network layers.

MANET characteristics such as mobility and power constraints add more complexity to the wireless medium issues. These features are also related to physical and Medium Access Control (MAC) layer characteristics. In several studies, researchers have shown that traditional routing metrics such as hop-count are not suitable for ad-hoc networks. Introduced by D. De Couto et al. at MIT, the idea that the ―shortest path is not enough‖ [12] has become a new paradigm spurring many researchers to introduce several new metrics for ad-hoc routing protocols. They believe that new metrics for MANET or mesh routing protocols should carry link quality or physical layer information.

Several metrics have been proposed to carry link quality measures in the backbone routing. ETX measures the number of successful packet deliveries as defined in [24], which is effectively used in selecting high-throughput paths. ETX is rendered ineffective if WMRs are configured with multiple interfaces, as shown in [5]. Since ETX finds links with low loss rates, in many cases, it ignores high bandwidth paths. For example, ETX tends to choose 802.11b, as it shows a lower loss rate than 802.11a, even though it provides much less bandwidth. Hence, two new metrics are proposed in [7]—ETT and WCETT—to find paths with higher throughput and lower interference.

There are several approaches to compute link quality metrics in the network layer, including packet count measurements [25] and cross-layer design with metric measurements at the physical layer and delivering to higher layers. In [6-8], new metrics such as interference and packet success estimation ratios are proposed that are communicated across the physical, MAC, and network layers. There are also other studies showing that QoS parameters could also be incorporated in the routing by using QoS metrics [22].

We propose link quality metrics for two types of paths between a source and destination MNs. Backbone (*bb_path*) and ad-hoc paths (*ah_path*) could show different link qualities with respect to each other. The characteristic differences between *bb_path* and *ah_path* suggest that a routing protocol

that embraces both paths should include separate metrics for each path.

WMNs have successfully overcome some of the ad-hoc network issues such as connectivity outage during hand-off, power shortage, and routing issues. Ad-hoc networks cannot use traditional routing protocols, mainly due to the ad-hoc characteristics mentioned above. However, WMNs do not suffer from those constraints. WMNs are characterized by fixed WMRs in the backbone that have unlimited power supply. Thus, theoretically, traditional protocols, with some modifications and improvements, could be used again. New solutions involving these ideas usually ignore ad-hoc constraints and try to improve routing performance in the backbone by introducing new metrics to the original protocols.

Routing proposals for the backbone have focused mainly on improving the current ad-hoc protocols by using multi-path options or new metrics that promise performance improvements. However, WMN has a major component that does not fall into the backbone. The access network in WMNs falls into the MANET, which carries characteristics of ad-hoc networks. In order to address routing in WMNs, we must clearly distinguish the characteristics of backbone and access and realize the fundamental differences between the two different parts of the network. WMN is comprised of a fixed backbone and mobile ad-hoc access sides. An integrated routing protocol that could address the needs of both networks should be aware of the path characteristics and take those into account while making routing decisions.

The authors in [9] propose MeshDV, a Mesh Distance Vector protocol, which takes into consideration both the backbone and the access sides of WMN. MeshDV combines proactive routing for the backbone with a reactive component for the client side. In MeshDV architecture, there is a client manager module that keeps two tables: a Local Client Table (LCTable) and a Foreign Client Table (FCTable). The LCTable holds information on all of the clients associated with a WMR, similar to MNs in our clusters, and a list of all WMRs that have inquired about the MNs. The FCTable holds information on all non-local clients and a pointer to their corresponding WMR. In their solution, WMRs perform all of the work and hold all of the information. Mobile nodes are not involved in routing decisions. The backbone is transparent to the mobile node. Like MeshDV, we also consider both backbone and ad-hoc access for routing. However, in our solution, the routing and decision-making is distributed between WMRs and MNs. We use a route table instead of an FCTable and do not need to keep routes from non-local clusters in the route table of each WMR. We also use a regular AODV cache table instead of an LCTable

Most proposed WMN routing solutions improve performance based on link quality solutions to overcome link failure. However, they do not address node-related issues such as node failure, medium access contention, and clusterhead congestion. Node failure or cluster congestion could potentially disconnect the corresponding cluster from the network. A comprehensive routing solution should address such issues as well. Our proposed solution will also address node-related issues by providing an *ah_path* that is completely independent of the WMRs and the backbone and could be used as a backup to the *bb_path* should a WMR fail or become unreachable.

2.3. Designing New Metrics for WMN

Designing an appropriate metric has major impact on the backbone routing. The shortest paths in wireless networks are not necessarily high throughput paths [12]. The ETX proved to be ineffective in cases where WMRs are configured with multiple interfaces [3], as in our case. Thus ETT and WCETT were proposed in [3]; both measure expected transmission time and can be used to find paths with higher throughput and lower interference. Reference [4] has introduced a framework for evaluating new WMN metrics. In their work, they show that WCETT addresses only intra-flow interference and is not isotonic (i.e., it cannot guarantee loop-free paths). Therefore, it is not a good choice for proactive link state and distance vector protocols. It is only good for on-demand protocols. In [4], the authors also propose a new metric called MIC (Metric of Interference and Channel switching), which favors paths that use less channel time. Hence, it takes into account inter-flow interference as well as intra-flow interference. It is discussed in [5] that hop count is still better than link quality metrics, such as ETX, WCETT, etc., for ad-hoc networks because frequent topology changes cause those metrics to recomputed link quality. The repetitive computations introduce significant delay and reduce throughout.

2.4. Mobile Nodes with Two Interfaces

MNs, like AMRs, could use two interfaces for communication with two networks, one interface to connect to the backbone AMR and the other to connect to other MNs in the ad-hoc network. MNs with two interfaces have become more popular in recent years, as they allow a user to connect to two separate networks simultaneously. MNs with one interface introduce several shortfalls in WMNs [26]. Using two interfaces has several advantages, such as ease in dealing with interference, enabling use of multiple radios, and enabling routers to connect multiple networks without causing interference and contention problems. However, when using one interface, if we need to switch the channel, we have to use channel switching and scheduling algorithms. Using multiple radios, routers have to deal with complicated algorithms for scheduling, and radio and channel assignment.

3. Wireless Mesh Network Architecture

WMN architecture is explained in detail in this section. We also discuss global connectivity and address components related to WMN, as well as how new metrics could help in routing the decision-making process of MN when it has to switch from a primary path (*bb_path*) to a back path (*ah_path*).

3.1. Backbone and Access Network Components

WMN architecture in this paper consists of WMRs in the backbone and clusters of MNs in the ad-hoc access network (Figure 1). Each MN is connected through an access link to an AMR, which serves as a gateway to the backbone network. Some WMRs in the backbone are connected to the Internet and serve as gateways to the Internet for the entire wireless mesh network. Those WMRs are called Internet Access Points (IAPs). Other WMRs closer to the access network are called Access Mesh Routers (AMRs). AMRs are the points of contact between MNs in the MANET and the backbone network. This architecture presents a three-layer structure, as illustrated in Figure 1.

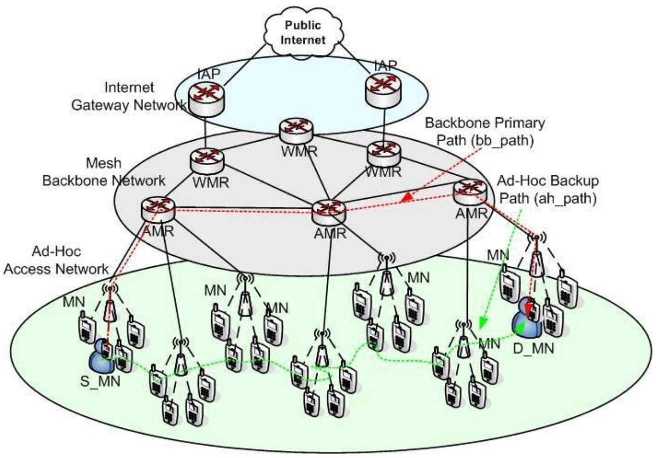

Figure 1: Architecture of the proposed WMN.

Each AMR has two 802.11 interfaces, the backbone interface (*bb_int*) and the access network interface (*an_int*). We use different radios for the *bb_int* and *an_int* to eliminate interference between the two paths. All *bb_int*s are equipped with 802.11a radios and connect AMRs to the backbone, whereas *an_int*s use 802.11b/g and connect AMRs to the MNs in the access network. Both *bb_int* and *an_int* are configured in 802.11 ad-hoc mode.

MNs are also equipped with two interfaces: an access network interface, called *an_int*, and an ad-hoc interface, called *ah_int*. MNs are connected with the backbone via AMR through *an_int*. They use their *ah_int* to form the ad-hoc network of MNs. Both interfaces can be implemented using 802.11b radios configured in ad-hoc modes on different channels. The *ah_int* of all the mobile nodes in the network are configured on a single channel to form the ad-hoc network. Use of MNs with two interfaces is discussed further in Section 4.4.

3.2. WMN Global Connectivity and Addressing

The mesh network consists of an IP network connected to the Internet via IAPs. A WMR may be connected with multiple mesh routers through the *bb_int*, creating multiple links. Each link requires a different IP subnet address as well. Hence, we create as many sub-interfaces (i.e. virtual interfaces) on a *bb_int* as the required number of subnets. The *an_int* forms the access link, which is assigned an IP subnet address as well. Thus, all of the MNs connected to the backbone through their access link receive an IP address on that subnet. MNs connected to the same WMR form a *cluster*, where the WMR becomes the *clusterhead* of that cluster or AMR. When an MN approaches the vicinity of an AMR, it receives the *an_int* beacon and connects to the AMR. If it moves from the coverage area of *an_int* of the old AMR to the new AMR, then it performs handover and changes its IP address by acquiring a new address on the subnet of the *an_int* of the new AMR. We allow the connectivity between an MN and its AMR through a multi-hop path composed of mobile nodes within the same cluster. Hence, a cluster of MNs and the associated AMR forms an ad-hoc network. The mobility at IP level can be managed by employing a variation of the IP mobility solution discussed in [10]. The mobility management in the proposed WMN is out of the scope of this study.

3.3. Routing in Wireless Mesh Network

The proposed routing scheme comprises integrated routing for WMN that considers the characteristics of both backbone and access networks. Between the S_MN and the D_MN, there are at least 2 paths: the *ah_path* and the *mesh_path*. For the *ah_path*, we use an AODV routing protocol. The *mesh_path* has 3 components: sub-path1 between S_MN and Source AMR (S_AMR), sub-path2 between S_AMR and Destination AMR (D_AMR), and, finally, sub-path3 between D_AMR and D_MN. Sub-path1 and sub-path3 are part of the *mesh_path*; however, they are access links and use AODV to establish the link (Figure 2).

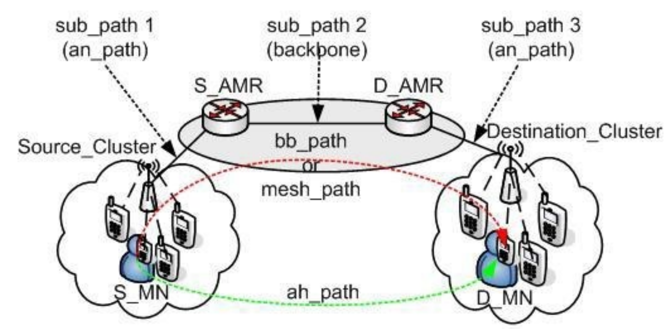

Figure 2: Access network; both *an_path* and *ah_path* use AODV.

We developed an extension to the AODV routing protocol that allows AMRs to act as a clusterhead, periodically send beacons to discover neighbors (i.e., MNs in their respective clusters), and proactively keep their local cluster's MNs in their AODV cache tables (or IP forwarding table). Thus, when an AMR receives a packet from another AMR, it will find the subnet and forward the packets to the corresponding AMR, continuing on the path to the destination.

If the D_MN is located in the same cluster as S_MN, then the route is discovered and packets are sent directly to the D_MN without going through the AMR. If the AMR receives a Route Request (RREQ) in which D_MN is in the same subnet as the S_AMR, the packet is dropped, assuming that there is a direct connection between the two MNs in the same cluster.

4. Proposed Backup Routing Design

The proposed integrated routing system for WMNs includes routing for the end-to-end path between source and destination MNs via two paths. In the backbone, several routing protocols have been proposed, such as AODV with different extensions, Dynamic Source Routing (DSR), Open Shortest Path First (OSPF), and so on. We designed OSPF in the backbone, as explained in Section 4.1, and AODV for the access and ad-hoc networks, as explained in Section 4.2. OSPF is a proactive and table-driven protocol, whereas AODV is an on-demand protocol. To the best of our knowledge, there has been no implementation for redistribution between these two protocols yet. Therefore, for the purpose of this study, we use OSPF and AODV for the backbone and access networks, respectively, and where necessary, we have provided routing information through extensions for AODV in the backbone. We study the routing system for MNs with one or two interfaces and allow the MN to choose the ad-hoc network over the backbone under constrained conditions in both cases. The new, modified AODV delivers the throughput information via Router Reply (RREP) packet to the MN to make the final decision on whether to take the *ah_path* or the *bb_path.*

4.1. Backbone Network Structure

OSPF is widely used in the Internet for intra-domain routing. It is a link-state routing protocol that requires every router to maintain a synchronized link-state database. The synchronization process involves the synchronization of link-state databases of two adjacent routers when they discover each other and the flooding of link state information throughout the network. OSPF improves the synchronization process by defining link types and limiting the scope of flooding. Both features cannot be directly implemented in the backbone of WMN. We propose schemes to implement them in the WMN backbone and give an outline of our proposal below.

In order to make the synchronization of adjacent routers more efficient, OSPF defines several link types such as point-to-point, broadcast, and non-broadcast multiple access (NBMA). It defines a Designated Router (DR) on a broadcast link to reduce the complexity of the n-squared adjacency problem [23]. Although a wireless link is a broadcast medium, due to the hidden node problem, neighboring nodes have a different set of neighbors in their transmission range, called the neighbor set. For instance, WMR-B in Figure 3 is connected to WMR-A and WMR-C through its backbone links, but A and C are not connected to each other through their backbone links, as they are outside of the transmission range of each other. This lack of consistency in the neighbor set of adjacent nodes due to the hidden node problem makes it difficult to elect a single DR. In our backbone design, we configure secondary interfaces to form separate broadcast networks. For example, two secondary interfaces can be configured on the single physical interface of B. A-B can be declared as a subnet on one secondary interface of B, while B-C can be declared as different subnet on the other secondary interface of B. We also designed a dynamic configuration algorithm for the assignment of subnets in the backbone network. The algorithm computes the neighbor sets for a node such that all of the nodes within a neighbor set are also neighbors to each other. We then assign a subnet to the neighbor set and configure secondary interfaces on all the nodes of the neighbor set. We use a heuristic to discover the maximal neighbor set by discovering a fully connected mesh of nodes.

OSPF allows a network to be structured as a hierarchy of areas, and it limits the scope of flooding of link-state information about the links inside the network within an area. It simplifies the hierarchy by restricting it to only two levels such that all areas are connected only through a single backbone area, called area 0. Configuring area 0 for OSPF in the WMN backbone may not always be simple. For instance, consider a WMN backbone as being deployed alongside a county road, stretched over many kilometers. In such a linear deployment, no central area exists that can be configured as area 0. Hence, we propose dividing the WMN backbone into autonomous OSPF networks that are connected through the Border Gateway Protocol (BGP). This is a novel use of BGP in a wireless network, which has never been proposed before that we are aware of. Although BGP in the Internet is known for unstable routing and long convergence time, most of its difficulties come from policy conflicts along the service provider boundaries. Since the WMN backbone is under a single administrative domain, inter-provider policy conflicts do not arise. A schematic representation of the backbone network design is presented in Figure 3.

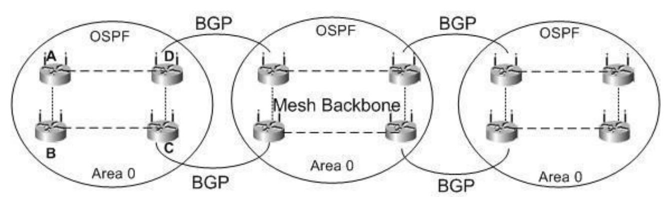

Figure 3: WMN backbone including OSPF area 0's. Areas are connected through BGP.

4.2. Access Network Structure

Consider paths between S_MN and D_MN in Figure 2. There are essentially two types of end-to-end paths: the *mesh_path* that traverses the backbone and the *ah_path* that goes through the MANET. The *mesh_path* has 3 segments: *sub-path1*, *sub-path2*, and *sub-path3*. Among the sub-paths, only *sub-path2* is composed entirely of links within the backbone; therefore, we call it the backbone path or *bb_path*. Sub-paths 1 and 3, called access network paths *(an_paths)*, are composed of access links, potentially multi-hop, formed within their respective clusters. Throughout this paper, we call *mesh_path* and *bb_path* interchangeably when comparing it to *ah_path*.

Generally, the paths within the backbone are more stable than *an_path*s because the WMRs are stationary nodes and the links among them are formed by directional antennas [11]. Dynamic link quality metrics such as ETX and WCETT can be

used in the backbone routing to perform multi-path routing within the backbone. However, the *an_path*s are the unstable segments of the *bb_path* due to channel contention, rate drops caused by increasing distance between an MN and the AMR, and instability due to node mobility. Hence, the *an_path*s could constrain the quality of a *bb_path* by, for example, lowering throughput or raising delay.

bb_path should be used as the *primary* path between a pair of source and destination MNs because of its tendency to traverse stable backbone links. The alternative *ah_path* is only used as a backup when *bb_path* is not available due to the conditions mentioned earlier.

An ad-hoc routing protocol such as AODV can be used to establish *an_path*s since the *an_path* is a part of the *bb_path*, which faces the contention problem that could become a bottleneck. On the other hand, the *ah_path* is a secondary path that should be set up only when required. Hence, for the *ah_path*, we also use the on-demand and ad-hoc routing protocol, which initiates route discovery only if required. AODV initiates route discovery when a new route is needed for packet forwarding or when an existing route is refreshed in the routing cache. The route discovery process typically involves the flooding of discovery packets inside the network, e.g., the flooding of RREQ packets in AODV. Since routes are not discovered or refreshed periodically in on-demand routing, it incurs less flooding overhead, which is suitable for a WMN ad-hoc access network.

4.3. Routing Model for Access and Backbone

Three routing decisions need to be made in order to solve the key issues in designing the proposed routing system. First, which node should decide on using either a primary or backup route? The route selection decision can be made either by the AMR or the MN itself. In either case, the *ah_path* is established by the MN. Hence, if the AMR makes the decision, then the information about the *ah_path* has to be transferred to the AMR, which necessitates discovering the full *ah_path* prior to making the decision. If the MN makes the decision, then it can delay the decision-making process until after the *ah_path* discovery. The MN can make the route selection in two steps. In the first step, it decides to initiate the route discovery based on the quality of the available *bb_path*. Then it can decide whether to use the primary or the alternative path after the full *ah_path* discovery with knowledge of the quality of the *ah_path*. Hence, we propose that the MN perform the route selection.

Second, when should the route discovery process for the *ah_path* be initiated? The *ah_path* route discovery is an expensive process; hence, we argue that it should be initiated only when there is a good chance of using the *ah_path*. We propose an algorithm for initiating route discovery in AODV, which is invoked by the mobile nodes. The source MN broadcasts AODV RREQ for the destination, setting the AODV RREQ-TTL = x, where x is the number of hops the MN is away from the AMR. When the AMR receives the RREQ from the source node, it checks the destination IP address. If the destination is in the local cluster, the AMR sends regular AODV RREP if it finds the route in its AODV cache. If the destination is not in the local cluster, the AMR will propagate the RREQ to the next hop and send the RREQ hop-by-hop to final destination. The D_MN will prepare a RREP packet that includes the throughput information as a new field and forwards the new RREP packet back to the source.

The third important issue in the design is how to decide between the quality of the *bb_path* and *ah_path*. The dynamic link quality metrics such as ETX and WCETT are effective measures of the throughput of backbone routes [2] and [3]. However, they are not as effective in an ad-hoc network [5]. A careful estimate of the round-trip time (RTT) of the *ah_path* could also be used as a measure of ad-hoc throughput. In our analysis, we used throughput as a performance measure. Each node has throughput information of its own link, which could be transferred to other nodes through backhaul transmission via piggybacking with control messages, or creating a special protocol for transmitting the throughput information. In the proposed routing system, we use piggybacking on the RREP message to deliver the throughput information back to the source. Using the RREP message to deliver the throughput information avoids incurring additional costs.

The design of an integrated routing protocol for WMN involves two major components: the first is the route discovery process in which MN finds the routes through both *mesh_path* and *ah_path*. In this situation, the MN evaluates the performance of the *mesh_path* and decides whether to use this path or to discover an alternative path through an ad-hoc network. The second component is path selection, which involves evaluating and comparing the route through *mesh_path* and *ah_path* and deciding when the backup path should be used.

4.3.1. Route Discovery Process

The S_MN broadcasts an AODV RREQ for D_MN. This RREQ could be captured by either another MN or by a WMR. The MN could be in the local cluster or in a remote cluster. The WMR could be the local clusterhead (AMR) or any other WMR along the way.

The route discovery procedures are implemented in Algorithms 1 and 2. Algorithm 1 passes hop count (hc) and throughput (Tput) parameters provided by the RREP message to Algorithm 2. Algorithm 2 will evaluate these parameters, and if they do not meet the threshold values (i.e. hc0 and Tput0), then it initiates a second route discovery, which is called every time hc or Tput falls below the threshold values. MN then waits to receive a RREP. Upon receiving RREP, MN checks to see if RREP is from an AMR or another MN. If it is from an AMR, then it should call the route discovery function. This function checks the hop count and throughput of the RREP, and if they fall below threshold, it initiates the second route discovery by sending a second RREQ; otherwise, it will enter the RREP into the route table. If the RREP is received from another MN, then it has to check whether the next hop of that MN is an AMR. In either case, the MN still calls the

second route discovery function. The difference is that if there is an AMR along the way, then the route type will be entered in the route table as *bb_path*.

Upon receiving an RREQ message, the MN checks the IP address of D_MN. If the D_MN is in the same subnet and the same cluster, the regular AODV procedure is used to resolve the route discovery. If the D_MN is not local but the route to D_MN is available, an RREP is sent to S_MN including the D_MN IP address, its hop count, and the throughput of the route. When an AMR receives the RREQ from S_MN, it checks the D_MN IP address; if the D_MN is in the local cluster, AMR uses the AODV cache and replies with a RREP, including the IP address of the destination, just as in regular AODV. If the D_MN is not in the local cluster, the AMR looks up the routing table. If it finds a route to the destination, it returns an RREP with the number of hops. A new field is added to the RREP packet format for *route_type*. *route_type* can hold the values *"bb"* (for *bb_path*) or *"ah"* (for *ah_path*). RREPs from the backbone are marked as *bb_path*, whereas RREPs from other MNs are marked as *ah_path*. A new column is also added to the AODV route table as *route_type*. Any route returned by the mesh router is entered in the route table as *bb_ path* or *ah_ path* depending on where it comes from. Once an RREP is sent by D_MN, it is tagged as *"ah."* At any stage, if it passes by an AMR or WMR, its *route_type* changes to *"bb"* and will remain *"bb"* until it reaches the S_MN. Therefore, if a RREP is tagged with *"ah"* for its *route_type* once it reaches S_MN, that means this route lies entirely within ad-hoc path, and there is no backbone router on this path.

```
Algorithm 1: Route discovery
Input: route reply control messages (RREP)
Output: second route discovery
Procedure:
    set hc0 = 3;
    set Tput0 = 0;
    broadcast RREQ;
    upon receiving RREP;
    if( route provider ip address == gw ip address)
        call algorithm 2 on ah_int
    elseif ( route provider ip address != gw ip address)
            if (NH == AMR)
                call algorithm 2 on ah_int
            else
                for (1 to hc)
                    if (rte_type == bb)
                        enter route as bb_path
                    elseif (rte_type == ah)
                        call algorithm 2 on ah_int
                        enter route as ah_path
                    endif
                endfor
            endif
    endif
    end
    Output: second route discovered
```

When the MN receives the RREP from the AMR, it decides whether the route provided by the AMR can satisfy the required threshold values set by Algorithm 2. If the required metrics fall below thresholds, then the MN should start a new route discovery by sending a second RREQ using AODV expanding ring search and finding a backup route through MANET.

Algorithm 2 sets the threshold values for throughput and hop count and collects the routing information. The S_MN compares the throughput value collected from the *bb_path* to the threshold values and decides whether to use the route provided by the AMR or to initiate a new route discovery.

```
Algorithm 2: Initiate route discovery
Input: hop count and throughput provided by RREP
Output: second route discovery request (RREQ)
Procedure:
    check hc
        if (hc < hc0) | (Tput < Tput0)
                initiate route discovery via ad-hoc
                (broadcasting RREQ with ttl = hc)
        else
                accept the route and enter hc in the route table
    end
    Output: broadcast second RREQ
```

S_MN initiates route discovery by broadcasting an RREQ to peer MNs and searching for a backup route within MANET. Upon receiving an RREP from ad-hoc network, S_MN enters the *route_type* as —*ah*‖ in the route table.

4.3.2. Path Selection Process

At this point, S_MN has performed a second route discovery and has two routes to choose from: *ah_path* and *bb_path*. This decision could be made using Algorithm 3. Algorithm 3 is a network-level implementation of the MN decision-making process.

```
Algorithm 3: Path selection
Input: two RREP control messages
Output: selected path with higher provided throughput
Procedure:
    route required
    check route table
    if no route available
        start algorithm 1
    else
        check throughput fields of RREP1 and RREP2
            set throughput = Tput_0
            get bb_path throughput = Tput_bb
            get ah_path throughput = Tput_ah
            d = ((Tput_ah – Tput_bb)/(Tput_ah))*100
                if (d > 25)
                    activate ah_path
                elseif (d <= 25)
                    activate bb_path
                endif
    endif
    end
    Output: Higher throughput path selected
```

MN only uses this algorithm if there are two routes available. It checks the route table; if there is no route, then it calls Algorithm 1 to find the routes. If there are 2 routes

available and it has to decide which one to take, then it checks the throughput provided by the two routes. Algorithm 3 calculates the threshold value of —*d*‖ by subtracting the two throughputs, dividing them by the *ah_path* throughput and multiplying by 100. —*d*‖ is a percentage value that determines the throughput difference between the two paths as a percentage value. Different network setups could assume different values for —*d*‖ depending on how reliable the backbone route is. For the purpose of this paper, we used a heuristic method to find an appropriate value for —*d*‖ that allows the path to change 1 out of 4 times. The assumption is that the *ah_path* is taken only if it provides a 25% higher throughput.

Algorithms 1-3 indicate that the MN uses *bb_path* until throughput falls below the threshold. When notified, the MN starts a second route discovery, finds the *ah_path*, and starts using this path if necessary. These algorithms ensure that the MN will switch to *ah_path* whenever throughput will fall below the threshold level. Such cases could happen when the MN is moving between clusters and there is latency, disconnection, or congestion.

In the proposed solution, the throughput is used as a performance measure. Each node has the throughput information of its own link, which could be transferred to other nodes through backhaul transmission via piggybacking with control messages, or by creating a special protocol for transmitting the throughput information. For the purpose of this paper, we rely on throughput measurements performed by OPNET.

4.4. MNs with 1 Versus 2 Interfaces

In this study, we have introduced, for the first time in WMNs, using MNs equipped with two interfaces: an access network interface (*an_int*) used to connect to the AMR and an ad-hoc interface (*ah_int*) used to connect to peer MNs in the *ah_path*. Both interfaces use the 802.11b/g radio; however, the *ah_int* is configured on a separate channel to connect to the *ah_int* of other MNs. Using MNs with only one interface poses several problems, as investigated in our previous studies [26]. For instance, in our solution, we introduce two different radios to be used for backbone and ad-hoc paths. Since we use an 802.11a in the backbone and an 802.11b/g on the access side for the *ah_path*, the AMR has to switch from *bb_int* to *an_int* once it redirects the traffic from the backbone to the access networks. However, for the MN to switch from the AMR connection to MN connection, it still stay on the 802.11b/g radio since both connections are on the access side, and they both use 802.11b/g. Two connections with two paths on the same interface and same radio would introduce performance degradation caused by contention and interference problems. Using two separate interfaces on the MN helps to alleviate these problems.

The other problem is that an MN with one interface in the intermediate clusters could communicate with either an AMR or another MN, but not with both at the same time on the same interface and the same channel. Therefore, if an MN is engaged in communication with the backbone, then it cannot respond to a communication request from another MN that has switched from the *bb_path* and is trying to start an *ah_path*.

Figure 4 shows how the MN with 2 interfaces could be connected to an AMR and another MN at the same time using two interfaces.

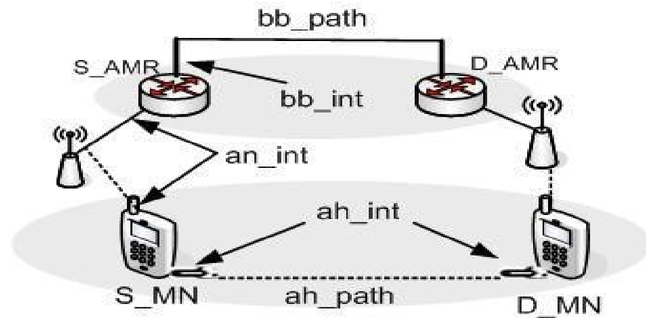

Figure 4: AMRs and MNs with 2 interfaces.

Using AMRs and MNs with 2 interfaces in the backbone, access and ad-hoc networks could introduce several backbone, inter-cluster, and intra-cluster interference issues. To eliminate these types of interferences, we implemented a careful channel assignment for both WMRs and MNs to carry multiple communications simultaneously using multiple non-interfering channels with several neighbors; such communications will not interfere with each other (Figure 5). Our interference-avoidance channel assignment eliminates backbone (Figure 5A), inter-cluster (Figure 5B), and intra-cluster (Figure 5C) interference.

Figure 5C shows how MNs use a separate channel called a common ad-hoc channel to carry all ad-hoc communications. All of the ad-hoc interfaces (*ah_int*'s) are assigned to this channel. Further interference-related discussions and channel assignment strategies are out of the scope of this paper due to space limitations.

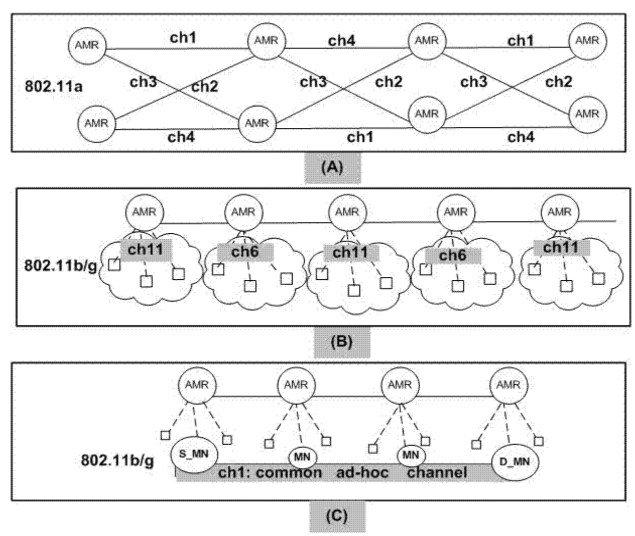

Figure 5: Channel assignment for WMN.
A) Alternating channel in the backbone.
B) Alternating channels in the access network.
C) MNs use a single channel for ad-hoc communication.

5. Performance Evaluation

The proposed routing solution is developed in a simulation environment implemented in the OPNET network simulation software [28]. The model is used to evaluate the performance of the proposed routing system. We run the simulation model in several scenarios under different conditions to test the routing capabilities of the newly developed WMRs and MNs in the proposed WMN both in the backbone and the ad-hoc networks.

5.1.Simulation Model

WMN is implemented in a simulation environment in OPNET modeler 14.5 PL1 [28] by creating three layers of network including the Internet access network, backbone mesh network, and ad-hoc access network. The three-layer equipment includes IAPs, WMRs and AMRs, and the MNs, respectively (Figure 6).

The access network includes clusters of MNs in a MANET structure, with AMRs as clusterheads. Each AMR is surrounded by a cluster of MNs. The first cluster on the left side is called the source cluster since it includes the S_MN, and the last cluster is destination cluster, which includes the D_MN. We used an 802.11a radio for the backbone and 802.11b/g for the access network. A campus network is deployed over a square geographical area of range 10*10 km^2, as shown in Figure 6.

The backbone network comprises WMRs in two rows. The lower row includes AMRs that connect MNs to the backbone. The upper row is core WMRs that participate in the backbone but do not have any MNs connecting to them for direct access purposes. The first AMR in the lower row is named S_AMR, which depicts the AMR corresponding to the source cluster, and the last AMR is D_AMR, which shows the AMR corresponding to the destination cluster.

WMN is deployed using two IAPs, four WMRs, four AMRs, and four clusters of MNs. Each AMR is surrounded by MNs in its cluster. For each cluster, we start the simulation with one MN and then increase number of MNs to start the effect of increased traffic and channel contention.

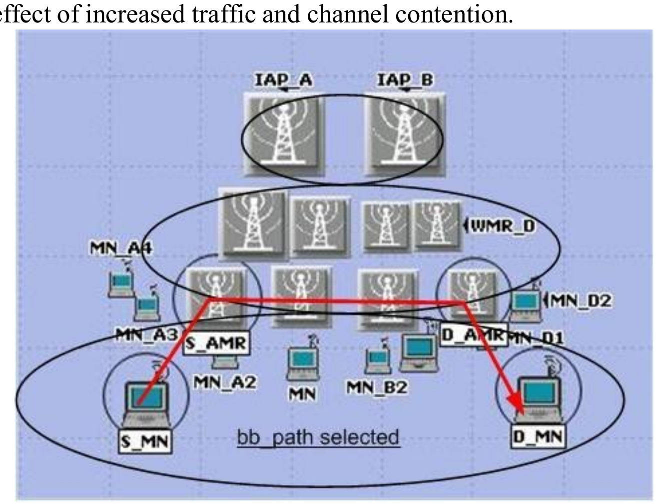

Figure 6: WMN from Figure 1, implemented in OPNET; *bb_path* selected by S_MN to D_MN.

The AMRs are equipped with two interfaces: one for the backbone (*bb_int*) running 802.11a and the other for the access network (*an_int*) running 802.11b/g, according to Figure 4. At the initial stage, MNs have a single interface running 802.11b/g to connect to both the AMR and the ad-hoc network. The assumption is that, initially, MNs use the same interface and the same radio frequency to connect to both the AMR and other MNs. This assumption is justified, considering that all nodes are in the ad-hoc mode and capable of connecting to more than one peer at the same time. *bb_int* is used for backbone communication with other peer AMRs or WMRs, and *an_int* used for access network communication with MNs in the cluster. Since the backbone is on 802.11a, backbone traffic will not interfere with MN-MN and MN-AMR traffic. At the second stage, we turn on the second interface of the MNs to be used for direct ad-hoc communication among peer MNs.

MANET traffic is generated between a pair of S_MN and D_MN using the traffic specifications shown in Table 1. Traffic is first generated from contending MNs in the cluster to go to the S_AMR. After 100 seconds, when the traffic is continuously generated and contention is stabilized, S_MN starts sending traffic to S_AMR. At this point, the new traffic is affected by the contention from other MNs.

The S_MN sends MANET traffic at exponential inter-arrival times of 0.01 seconds, and the constant packet sizes are 8,192 bits for the D_MN and 16,384 bits for the AMR. We set the throughput threshold at a minimum value of 100 bits/sec in order for the second route discovery to be triggered. The simulation ran for 4 minutes each time, and it is repeated 10 times for each experiment. Setting the seed number option of OPNET on 20 in each experiment provides an average result equivalent to 200 times in each case.

Table 1: Traffic parameters generated from S_MN to D_MN

Traffic Parameter	Value
Start time	100 (0 sec for contending MNs)
Inter-arrival time	0.01 sec
Packet size	8192 or 16384 bits (depending on destination)
Destination	D_MN, (AMR for contending MNs)
Stop time	End of simulation

In the following sections, several scenarios are presented with AODV, including throughput and delay results. In each case, the results are presented using the Cumulative Distribution Function (CDF) for both throughput and delay analysis. CDF function is used since in the selected simulation environment, the performance measures are cumulative, and the CDF shows a clear indication of the collective performance measures over time.

5.2.Basic Topology, Including Backbone and Access

The results for the scenario in Figure 6 are presented in Figure 7. The throughput results for *bb_path* are presented for three

different channel contention situations. Link throughput is measured at the destination node. We increase channel contention by increasing the number of MNs in the source cluster from two to six.

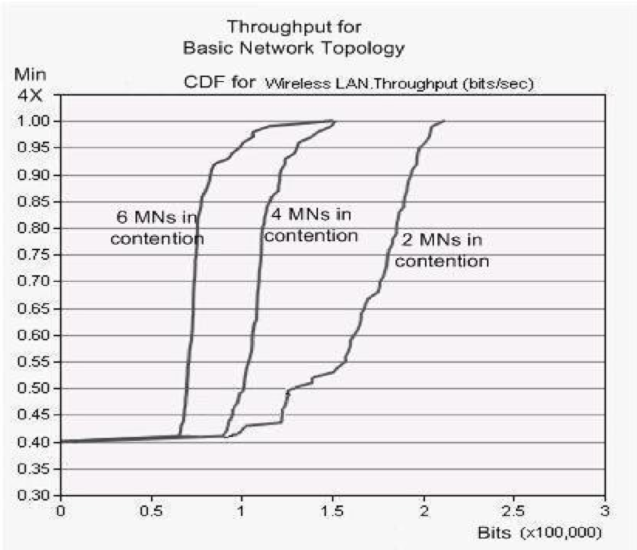

Figure 7: Throughput at the destination while increasing number of mobile nodes in a cluster resulting in increasing contention.

We observe from Figure 7 that the throughput at the destination MN decreases from over 200 Kbps to almost 60 Kbps, while the numbers of MNs in the source cluster increases from two to six. This is due to contention surge as number of MN increases at the source cluster, and, consequently, the packet drop rate will increase. This is verified by measuring the number of retransmissions in the source clusters, which also increases with the decrease in throughput. It illustrates the situation when high contention in the source cluster renders *an_path* to be the bottleneck of *mesh_path*.

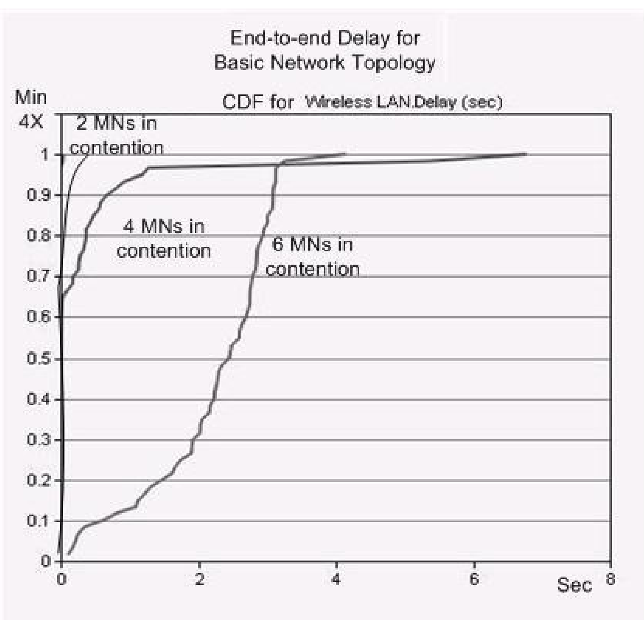

Figure 8: End-to-end delay while increasing number of MNs in a cluster resulting in increasing contention.

These results could also be confirmed with the end-to-end delay between S_MN and D_MN for 2 versus 6 MNs in the source cluster, as illustrated in Figure 8. It shows that the delay will rise dramatically as the number of MNs increases in the source cluster. This is clearly due to the contention level increase in the source cluster.

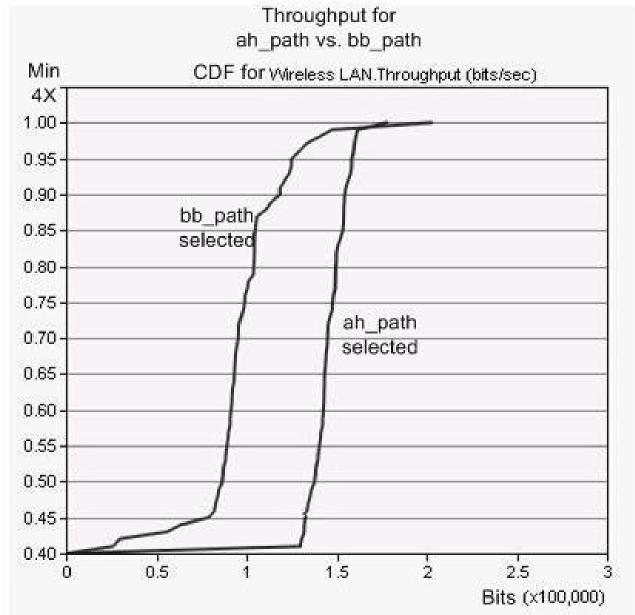

Figure 9: Throughput at the destination comparing *bb_path* versus *ah_path* for the case of high contention in the source cluster.

Figure 9 shows the scenario with 4 MNs, where we allow traffic to pass through the backbone or ad-hoc paths individually and measure throughput for each case separately. This figure shows clearly that *ah_path* could improve performance when the *bb_path* is constrained by contention for over 40%.

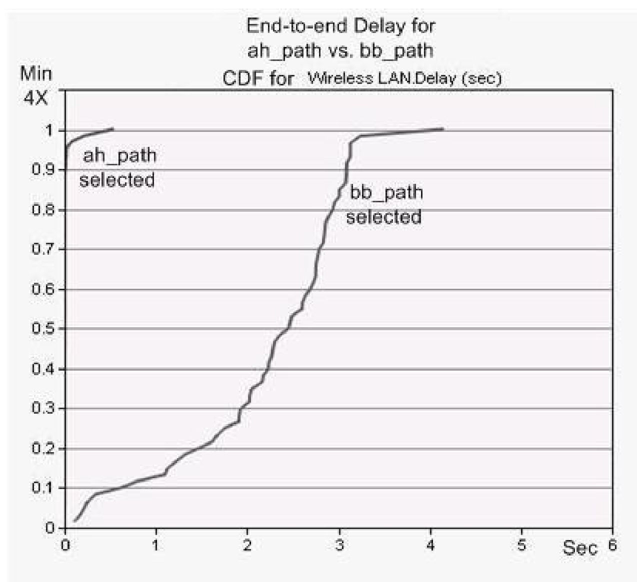

Figure 10: End-to-end delay comparing *bb_path* versus *ah_path* for the case of high contention in the source cluster.

The corresponding delay results in Figure 10 clearly show that the system could decrease the delay significantly if the MN chooses to take the alternative *ah_path* over the congested *bb_path*. Figure 10 shows that the delay is almost diminished when the MN switches from *bb_path* to *ah_path*.

5.3.WMN Topology with the New Routing Scheme

The performance of *bb_path* and *ah_path* are evaluated using the throughput measurements provided by the AODV RREP messages, which in turn help in routing decisions and path selection processes. The evaluation and decision-making processes are developed in the proposed algorithms and implemented in new scenarios.

This part of the simulation is based on the changes in the core of AODV source code in OPNET. The new AODV_enabled nodes should be aware of the throughput values for each path. Each AMR measures its link throughput to the next hop or next AMR (this value is saved as *own_throughput*).

Based on the current implementation, S_MN broadcasts the RREQ. S_AMR receives the RREQ and uses regular AODV to forward it hop by hop to the destination. D_MN replies with a unicast RREP message back to the source including link throughput. This is a one-way downlink throughput of D_AMR to D_MN, not the throughput for the reverse path. D_MN also sets *route_type* to *"ah."* D_AMR receives the RREP, compares its throughput (recorded as intermediate_throughput) with its own throughput, and updates the RREP throughput with the smaller value. Every AMR along the way compares this throughput with its own link throughput and updates the RREP with the smaller value. Since the throughput provided by the backbone links are usually higher than any access network throughput, the original link throughput coming from D_MN, which represents the throughput of sub_path3, is likely smaller than any backbone link throughput and likely to be selected as the path throughput of *mesh_path*. Therefore, this throughput will have to compete with the throughput of sub_path1, and the smaller value of the two will get elected as the throughput for the route. At the same time, D_AMR will also change the *route_type* to bb, which remains the same for the rest of the journey back to the source.

If the RREP throughput is less than the threshold throughput and the second route discovery is initiated, a second RREQ will go through *ah_path* to the next MN and use regular AODV to travel hop by hop to the destination. Thus, D_MN will have a second RREQ from *ah_path*. D_MN will send a second RREP through *ah_path*, and a procedure similar to the one in the *bb_path* will be repeated, except that *route_type* will always remain *"ah"* for this path. The throughput added to RREP on the ad-hoc path is the link throughput between the D_MN and the next hop (neighboring MN). Each MN along the way will compare this throughput with its own link throughput to the next MN and update the RREP accordingly.

At this point S_MN will have two routes—*"bb"* and *"ah"* — with each having its own throughput value. S_MN will compare these two throughput values and use the equation in Algorithm 3 to decide which path to select. The AODV routing tables include two new columns for *route_throughput* and *route_type*. The value of *route_throughput* could be the value of throughput collected from the RREP message for —*bb_path"* or *"ah_path"* depending on whether the last node is an AMR or MN, respectively. The value of *route_type* is a Boolean value (*"bb"* or *"ah"* for AMR or MN, respectively). This is determined by extracting the last digit of the IP address of the source in the RREP. The AMRs are clusterheads, and their IP addresses are statically set to x.x.x.1; therefore, if the last digit of the IP address is 1, then the source is an AMR and the *route_type* is set to *"bb"*; otherwise, it is set to *"ah."*

The new AODV source code includes the throughput value in the routing cache and RREP packet and is implemented in the OPNET module. Then the new source code is compiled and the simulation ran for each scenario separately. Once the MN receives the RREP packet, it is informed of the throughput values for the backbone, and it does a comparison with a threshold value for throughput. If the RREP-reported throughput does not meet a minimum requirement set by the threshold, then MN will switch to *ah_path*.

The results for the scenarios with the new source code are presented in Figures 11 through 17. Figure 11 shows the throughput results for the basic scenario by increasing the number of MNs from two to six. In the presence of 2 MNs in the source cluster, *bb_path* is selected. By increasing the number of MNs in the source cluster from 2 to 4, S_MN still chooses the *bb_path*; however, the throughput drops by almost 40%.

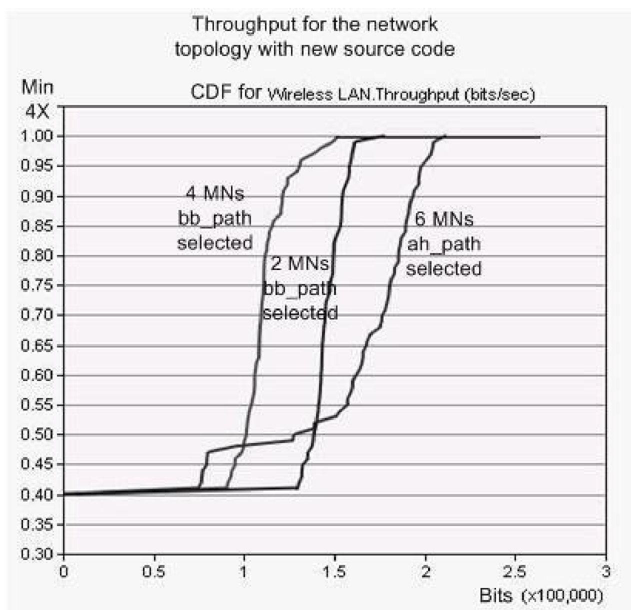

Figure 11: In the presence of 6 MNs, throughput at the destination drops, MN switches path to ad_path to compensate, returning throughput to that of 2MN.

In raising the number of MNs in the source cluster to 6, the trend suddenly changes. We observe in Figure 11 that the throughput in the presence of 6 MNs has increased in comparison to selecting *bb_path* with 4 MNs. Initially, there is a small drop in throughput to about 60% of the case for 2 MNs. Then we observe a surge of over 50% to almost 150 Kbps. This indicates a switch from *bb_path* to *ah_path* quickly after the start time. The increase is similar to that observed in Figure 9. The improved performance surpasses that of 2 MNs.

The delay performance measurements illustrated in Figure 12 show a clearer picture of the results. As illustrated in Figure 12, delay increases significantly from 2 to 4 MNs in the source cluster while using *bb_path*. However, when the S_MN chooses the *ah_path* as an alternative path in the presence of 6 MNs, the delay drops significantly to a level below that of 2 MNs.

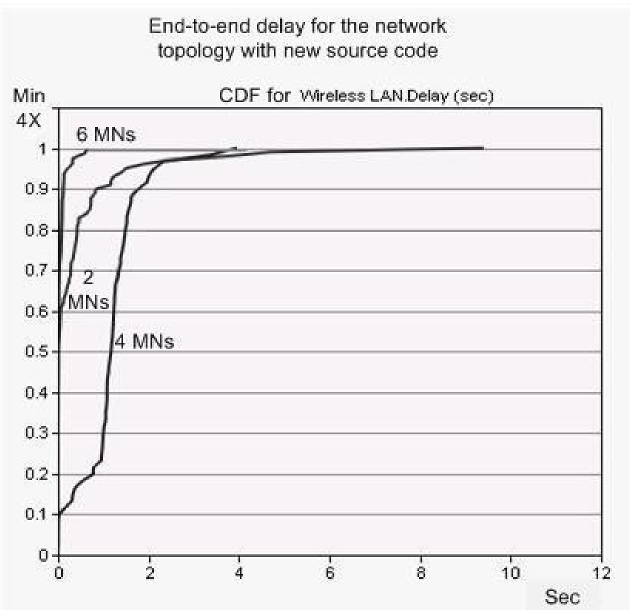

Figure 12: In the presence of 6 MNs, end-to-end delay drops dramatically almost to that of 2MNs for the basic case.

By increasing the number of MNs from 2 to 4, the delay is increased due to increased contention in *bb_path*. However, when the number of MNs is increased further to 6, the delay reduces dramatically to almost zero until the very end of the simulation. This clearly indicates that the *ad_path* is selected, and it has a great effect on the delay.

Figure 13 shows the actual OPNET network topology for the scenario with 6 MNs in the source cluster. S_MN favors the *ah_path* due to the fact that throughput performance is decreased below the minimum requirement set by Algorithm 2.

Figure 14 shows the throughput performance for the case in which S_MN chooses *ah_path* over *bb_path* in the presence of 6 MNs. As the number of MNs in the source cluster increases to 6, the throughput decreases initially to a level lower than that of 4 MNs to about 70 Kbps. This indicates that the traffic in the presence of 6 MNs initially uses the *bb_path*. Eventually, S_MN will switch from *bb_path* to *ah_path* due to its higher throughput available.

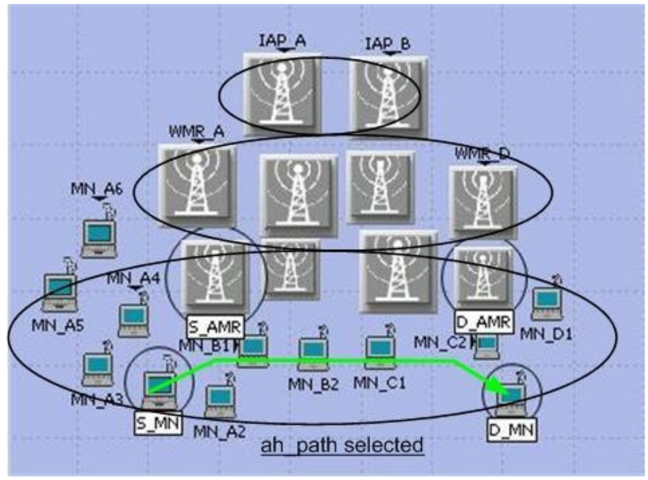

Figure 13: WMN with new AODV source code, 6 MNs in the source cluster, and *ah_path* selected.

The throughput in the presence of 6 MNs increases in comparison to selecting *bb_path* with 4 MNs. The drop in the throughput is due to the fact that, initially, the next hop node for S_MN is still S_AMR, and S_MN still sends traffic via backbone. At this point, there are still 6 MNs contending for the channel (contention level is 6).

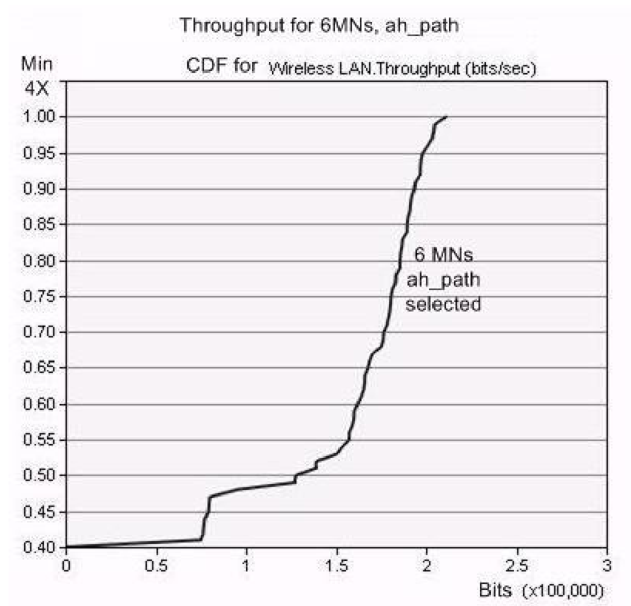

Figure 14: Throughput values for scenario in Figure 10 with 6 MNs. S_MN will switch from *bb_path* to *ah_path.*

After about 120 seconds, we observe improvement in throughput. This is due to the initial surge when the switch to the *ah_path* takes place. At this point, traffic is switched and starts traversing via the *ah_path* and, consequently, the throughput increases to the throughput close to that of 2 MNs and constantly increases until it reaches around 200 seconds. After the initial switching surge, the throughput starts stabilizing at a point that sits between the throughput of 2 MNs and 4 MNs scenarios and continues at a steady rate beyond this point.

5.4. Routing Performance using MNs with 1 Versus 2 Interfaces

We created MNs with two interfaces in OPNET and rebuilt the scenarios using the new type of MNs. MNs with two interfaces could carry simultaneous communications with both backbone routers and other peer MNs. Specifically for MNs in the middle clusters that are already engaged in a backbone communication with their own AMR, it would be easier to accept new calls from peer MNs using their new interface dedicated for ad-hoc communication. We set all of the simulation conditions and parameters as in the previous scenarios and ran the simulations to compare the performance of the routing scheme using one versus two interfaces. Figure 15 shows the throughput results for MNs with one versus two interfaces.

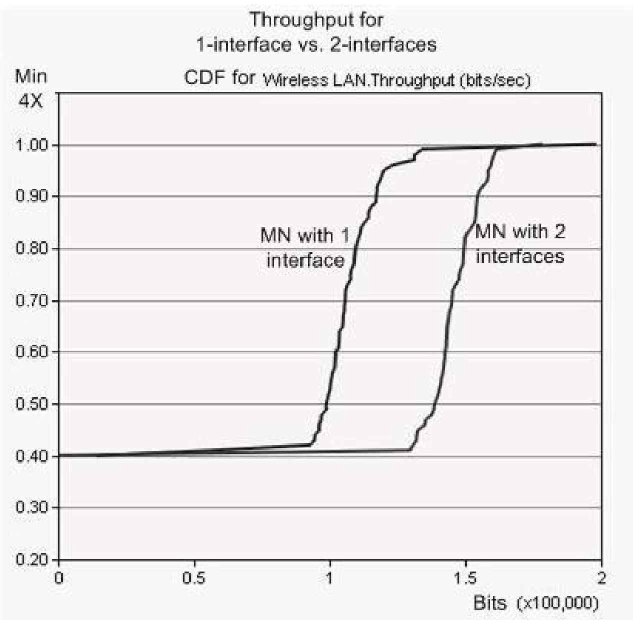

Figure 15: Throughput values for WMNs including MNs with 1 interface versus 2 interfaces.

Figure 15 shows that the overall performance of MNs with two interfaces is higher than that of MNs with one interface. During the course of simulation, the throughput is improved for both cases, but it is much faster in the case of MNs with two interfaces. MNs with one interface in the middle clusters will have multiple connections with WMRs and MNs and have to switch from *an_int* to *ah_int*, when the ad-hoc communication starts. During these operations, contention arises and throughput improvement is impaired. However, in the case of MNs with two interfaces, the throughput improvement is steady throughout the simulation.

The results could be observed more clearly by looking at the delay performance measurements illustrated in Figure 16. As illustrated in Figure 16, delay decreases significantly from 1 interface to 2 interfaces. It is clearly observed that in the presence of two interfaces on the MNs, the packets choosing to go through the *ah_path* do not need to wait for the path switch and could immediately switch to *ah_int* and select the ad-hoc specific channel to go through. Therefore, the delay is close to zero.

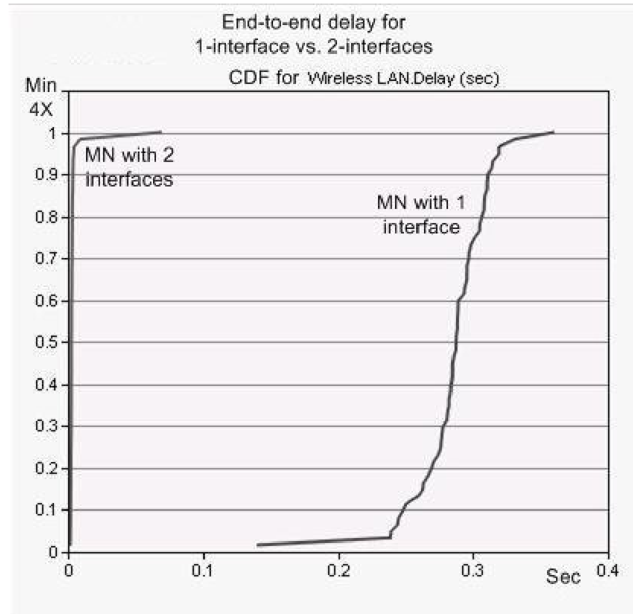

Figure 16: End-to-end delay values for WMNs including MNs with 1 interface versus 2 interfaces.

To further investigate the effect of two interfaces on the MNs, we also looked at the system throughput. A big impact that could result from using a second interface is eliminating interferences between the backbone and ad-hoc communications. We expect that this will result in a major improvement in the overall system throughput.

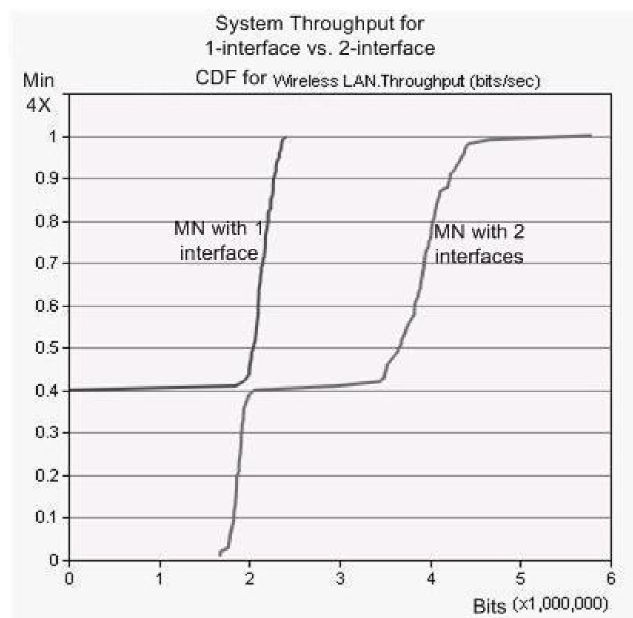

Figure 17: System Tput values for WMNs with 6 MNs. MNs with 1 interface versus 2 interfaces

It is observed in Figure 17 that the effect of using MNs with two interfaces could dramatically improve the overall

performance of the network. The total system throughput has increased to almost twofold.

6. Conclusions and Future Work

We propose an integrated routing system for a WMN that exploits both paths through the backbone and ad-hoc access networks. The motivation for this research study is to consider *ah_path*as an alternative or backup path to be used under critical conditions when *bb_path* is not available or severely constrained. We have simulated the access contention situation and demonstrated the benefit of alternative *ah_ path*. We also proposed a scheme for initiating the route discovery and path selection of the ad-hoc path.

We incorporated throughput information in the route cache and RREP packet of AODV and allowed AODV to inform MN of the throughput information in addition to the regular hop count. We also enabled MN to make a routing decision based on the throughput information.

We created MNs in OPNET with two interfaces and compared the results with those of MNs with one interface. Overall, the MNs with two interfaces show higher improvement in throughput and significantly lower delay during the course of simulation. In future works, we will create similar MNs with two interfaces and build more scenarios to further investigate these results.

In the future, we also plan to incorporate other link quality metrics (e.g. ETX) in AODV. We also want to incorporate QoS metrics in the decision of using *ah_path*. We are developing a routing-based framework for mobility management in WMNs that will use *ah_path* to hide the handover-related losses and delay.

References

[1]F. Akyildiz, I. Wang, and X. Wang, Wireless Mesh Networks: A Survey, Computer Networks Journal (Elsevier), March 2005.

[2]R. Baia, M. Singhala, Y. Wanga. ―On supporting high-throughput routing metrics in on-demand routing protocols for multi-hop wireless networks‖ Journal of Parallel and Distributed Computing, Volume 67, Issue 10, October 2007, Pages 1108-1118

[3]R. Draves, J. Padhye, and B. Zill, Routing in Multi-radio, Multi-hop Wireless Mesh Networks, ACM MobiCom, Philadelphia, PA, September 2004

[4]Y. Yang, J. Wang, R. Kravets; Designing Routing Metrics for Mesh Networks. WiMesh-2005, Santa Clara, CA 26th Sep 2005.

[5]R. Draves, J. Padhye, and B. Zill, Comparison of Routing Metrics for Static Multi-Hop Wireless Networks, SIGCOMM'04, Aug 2004,Portland, USA.

[6]L. Qin and T. Kunz, survey on MANET routing protocols and cross layer design, Technical Report SCE-04-14, Carleton University.

[7]L Iannone, et al, Cross-layer Routing in Wireless Mesh Networks, 1st Int. Symp. in Wireless Communication Systems, Mauritius, September 2005.

[8]R. d. Renecsse, et al, QoS enabled Routing in Mobile Ad-hoc Networks. Technical Report. Electrical Engineering Department, Centre for Telecommunications Research, King's College London, UK.

[9]L. Iannone and S.Fdida, MeshDV: A Distance Vector mobility-tolerant routing protocol for Wireless Mesh Networks, IEEE REALMAN, 2007

[10]J. Wu, M. Jaseemuddin, and A. Esmailpour, Integrating UMTS and Ad Hoc Networks, Proceedings of IEEE Int. Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2006), Montreal, Canada, August, 2006.

[11] Nortel network publication on-line. Available on the Internet in December 2009 at: www.nortel.com/solutions/wrlsmesh/index.html

[12]D. De Couto et al, Performance of Multi-hop Wireless Networks; Shortest Path is not enough, the 1st Workshop on Hot topics in Networking (HotNets- I), 2002.

[13]E.M. Royer and C-K Toh, ―A review of current routing protocols for MANET‖, IEEE Personal Communications, vol. 6 No 2, pp. 46-55. April 1999.

[14]S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communication Networks, Oct. 1996, pp. 183-97.

[15]Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In Proc. ACM/IEEE MobiCom, pages 85–97, October 1998.

[16]DSDV (Highly Dynamic Destination-Sequenced Distance Vector routing protocol) - C. E. PERKINS, P. BHAGWAT Highly Dynamic Destination-Sequenced Distance Vector (DTDV) for Mobile Computers Proc. of the SIGCOMM 1994 Conference on Communications Architectures, Protocols and Applications, Aug 1994, pp 234-244.

[17]P. JACQUET, P. MUHLETHALER, A. QAYYUM, A. LAOUITI, L. VIENNOT, T. CLAUSEN. OLSR: Optimized Link State Routing Protocol , RFC 3626, http://www.olsr.org/

[18]Ad-hoc On-demand Distance Vector - C. PERKINS, E.ROYER AND S. DAS Ad hoc On-demand Distance Vector (AODV) Routing, RFC 3561

[19]D. JOHNSON, D. MALTZ, Y. HU: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks, Internet Draft, draft-ietf-manet-dsr-10.txt, work in progress, July 2004

[20]S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, ―Cross-layer design for wireless networks.‖ WNCG Technical, Report TR-2003-04-00001, Wireless Networking and Communications Group, Department of Electrical and Computer Engineering, The University of Texas at Austin. Available for download at http://www.wncg.org. 2009

[21]M. Chiang, To layer or not to layer: balancing transport and physical layers in wireless multihop networks, in: IEEE Annual Conference on Computer Communications (INFOCOM), 2004, pp. 2525–2536.

[22]R. d. Renecsse, et al, QoS enabled Routing in Mobile Ad-hoc Networks. Electrical Engineering Department, Centre for Telecommunications Research, King's College London, UK.

[23]J. Moy, OSPF-Version 2, RFC 2328, April 1998

[24]D. S. J. De Couto, et al, A High-Throughput Path Metric for Multi-Hop Wireless Routing, Wireless Networks 11, 419–434, 2005.

[25]Srinivasan Keshav, ―A Control-Theoretic Approach to Flow Control‖, ACM Annual International Conference on Mobile Computing and Networking (MOBICOM), 1991

[26]A. Esmailpour, M. Jaseemuddin, N. Nasser, and O. Bazan. ―Ad-hoc Path: an Alternative to Backbone for Wireless Mesh Networks‖. Proceedings of IEEE International Conference on Communications (ICC), Glasgow, U.K., 24-28 June, 2007.

[27]U.C. Kozat, I. Koutsopoulos, L. Tassiulas, A framework for cross-layer design of energy-efficient communication with QoS provisioning in multi- hop wireless networks, in: IEEE Annual Conference on Computer Communications (INFOCOM), 2004, pp. 1446–1456.

[28]OPNET technologies Network R&D solutions, OPNET modeler, available on Internet, 2010, at: http://www.opnet.com/solutions/network_rd/modeler.html