Development and Performance Evaluation of Ad-hoc Wireless Test bed

[1]N.Pradeepa Kumari, [2]D. Vineela, [3]N.Kasiviswanath, [4]S.Zahoor-ul-Haq

[1,2] *Research Scholar in IITH*
[1]pradeepa.kumari@gmail.com
[2]vineela1289@gmail.com

[3, 4] *Department of CSE,*
G.Pulla Reddy Engineering College
[3]hodcse@gprec.ac.in
[4]s_zahoor_2000@yahoo.com

Abstract - **Performance evaluation of multi-hop Ad-hoc network routing protocols have been primarily conducted through analytic and simulation-based studies, which frequently fail to accurately predict real-world performance and behavior. The main reason for this challenge is developing test bed with ALIX boards, which are cheaper and roust embedded devices. We have developed Ad-hoc connection in indoor environment using a number of ALIX boards and two laptops using OLSR protocol. We successfully transferred the video file for different hops in different scenarios and evaluated the performance from that. However though the link quality is better the file is received with more error rate, and though the network topology is stable the link breaks.**

*Keywords***: Ad-hoc, ALIX Boards, Voyage OS, Loss, Delay**

I. INTRODUCTION

A multi-hop Ad-hoc network is an autonomous system, composed of series of wireless nodes which can communicate with each other through multiple hops, even if the source node's wireless signal cannot reach the target node. Nodes in such a network are working both as terminal and wireless router. And specially, if to emphasize node mobility, we get the mobile Ad-hoc networks (MANET [1]). A mobile Ad-hoc network (MANET) is a dynamic, multi-hop wireless network established across a collection of mobile nodes on a common wireless channel. MANETs differ from traditional wireless networks by enabling nodes to communicate without requiring a centralized infrastructure such as a wireless access point. This emerging technology is often proposed as a next generation communications system to provide robust, flexible and versatile wireless communication in the absence of functional infrastructure.

Since MANETs are multi-hop wireless networks without a pre-installed infrastructure, they can be deployed instantly in situations where infrastructure is unavailable (e.g., disaster recovery), or where infrastructure is difficult to install (e.g., battle fields). It is maturing as a means to provide ubiquitous unthread communication. With the increase both in the bandwidth of the wireless channels and in the computing power of mobile devices, it is expected that video service will be offered over Ad-hoc networks in the near future. (1) In disaster recovery, e.g., firefighters carrying cameras and wireless transceivers can track each other's where aborts and send the fire scene back to the command center. (2) A sensor network can be deployed to monitor wildlife in an inaccessible region, where live video is captured and relayed back to the base by the sensors scattered in the region.

Ad-hoc networks pose a great challenge to video transport. There is no fixed infrastructure and the topology is changing frequently due to node mobility. Therefore links are continuously established and broken. The availability and quality of a link further fluctuates due to channel fading and interference from other transmitting users. In addition, an end-to-end path consists of a number of wireless links. Thus transmission loss in Ad-hoc networks is more frequent than in wireless networks with single hop wireless paths connecting nodes to the wire line infrastructure.

OLSR is a proactive protocol adapted to the mobility of the network. The simulation results of multi-path routing protocols offer generally better performance compared with the single path protocols [16]. The test bed also resulted the same.

In this paper, we gave clear description about the ALIX boards and CF cards. And also the clear description about system design along with the used commands. Section II presents an overview and operation of the Optimized Link State Routing Protocol (OLSR), which is used for establishing multi-hop Ad-hoc routes; Section III clearly describes the system design, which includes the description of ALIX boards, CF cards, Voyage Linux OS, Ad-hoc connection, OLSR installation and clock synchronization; Section IV describes the test bed implementation and Section V presents the test bed results discusses its performance over each scenario; finally section VI presents the conclusions of the paper.

II. OLSR PROTOCOL OVERVIEW

There are four main protocols currently being considered for standardization by the IETF MANET group: Ad-hoc On Demand Distance Vector (AODV) [2], Dynamic Source Routing (DSR) [4], Optimal Link State Routing (OLSR) [3], and Topology Broadcast Based on Reverse-Path Forwarding (TBRPF) [5]. We chose OLSR as the routing protocol in our test bed.

The Optimized Link State Routing protocol (OLSR) is a routing protocol that is optimized for mobile ad-hoc networks but can also be used on other wireless ad-hoc networks. It is a proactive link-state routing protocol that

floods a topology table of its neighbors to all nodes in the network which then compute optimal forwarding paths locally.

A. Operation of OLSR

In OLSR protocol, there are four main steps to create a route table. Neighbor sensing, MPR (Multi-Point Relay) selection, MPR information declaration and route table calculation. During the neighbor sensing, each node periodically broadcasts the HELLO message, containing information about its one-hop neighbors and their link status. The HELLO messages are received by all the one-hop neighbors, but they are not retransmitted to further nodes. Upon receipt of the HELLO messages, each node can update its knowledge of its one-hop neighbors and two-hop neighbors. This information is recorded in a neighbor table. On the basis of this information, each node performs the selection of its MPR set. During the MPR selection, each node independently selects its MPR set according to the MPR selection scheme.

As a result, all of the two-hop neighbors of each node are contained in the union of the neighbor sets of its MPRs. Then, each node declares its MPRs in the subsequent HELLO messages. From the HELLO messages, which contain the MPRs, each node can inform its MPR selectors and construct its MPR Selector table. During the MPR information declaration, each node broadcasts specific control messages called Topology Control (TC) messages to declare its MPR Selector set. The TC messages are forwarded through MPR nodes and transmitted to all nodes in the MANET. According to the MPR selectors and the information in TC messages, a node maintains a network topology table to record the MPRs of other nodes. The topology table is a base of calculating the route table. During the route table calculation, a node calculates the route table based on the information contained in the neighbor table and the topology table. To find a path from a source (S) to a destination (D), an intermediate node (I1) one hop away to D has to be found and the connection pair [I1, D] obtained. Then a node (I2) one hop away to I1 has to be found and te connection pair [I2, I1] obtained. And so forth, until a node In is found in the MPR sets of S. Based on this process, the route table is built by tracking the connection pairs included in the topology entries in a topology table.

III. System design

The aim of our experimental study was to investigate the performance of multi-hop ad hoc networks in indoor scenarios. A number embedded devices ALIX 3D3 boards and laptops were used to facilitate this test bed.
The Compact Flash (CF) cards, containing Voyage-0.8.0 Linux OS [6], which include 2.6.32 kernel version, are fixed to ALIX Boards. The PC Engines ALIX series of system boards gives OEMs a higher performance replacement for the WRAP series of system boards. Fig 1 shows the ALIX board that we used for our project. Voyage Linux is a Debi an derived distribution that is best run on a X86 embedded platforms such as PC Engines ALIX/WRAP, Soekris 45xx/48xx/65xx and Atom-based boards. It can also run on low-end x86 PC platforms. Typical installation requires 128MB disk space, although

larger storage allows more packages to be installed. Voyage Linux is so small that it is best suitable for running a full-feature firewall, wireless access point, Asterisk/VoIP gateway, music player or network storage device. And Laptops were installed with Ubuntu 11.10 version OS, as it is a open source and it is so simple to download software packages that you requires and easy to remove those that are adding unnecessary burden through unneeded processing.

Fig1. ALIX 3D3 with 8GB CF card

A.Ad-hoc Network Creation

Once the ALIX Boards and laptops are ready, those should be configured to Ad-hoc mode to communicate each other. To set the nodes to Ad-hoc mode, [8] ensure that network manager has been stopped and then bring the interface down by issuing the commands *sudo service network-manager stop* and *sudo ip link set wlan0 down* in terminal. Then switch the card to ad-hoc mode using *sudo iwconfig wlan0 mode Ad-hoc* and name(ssid) for the network you want to create using *sudo iwconfig wlan0 essid "adhoc"* commands. Then bring the interface back up using *sudo ip link set wlan0 up*. Assign the IP, broadcast and netmask addresses using *ifconfig wlan0 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255*. Start the network manager using *sudo service network-manager start* Make sure that all nodes are to be have the same channel. Currently Network Manager does not support static address settings and Network-Admin does not support Ad-hoc networks. So instead we can configure the network in the */etc/network/interfaces* file. Node IP addresses were selected from 192.68.1.x range. Now nodes are ready with ad-hoc configuration, so that they can directly communicate each other. Using an ICMP echo message *ping* is used to test the connectivity between the nodes. If the nodes are communicating with each other means they are in Single-path Ad-hoc mode.

B. OLSRd Installation

The newest stable version *olsrd-0.6.2* [7] was used to facilitate this test bed. The tar file is downloaded and extracted. Unpack, compile and install the source code [9]. To compile navigate to *olsrd-0.6.2* directory and issue the commands *sudo make* and *sudo make install* for compilation. Make sure that the packages *Bison* and *Flex* should be installed before installing olsrd. Otherwise it asks to install those. *sudo apt-get install bison* and *sudo apt-get*

install flex commands are used to install Bison and Flex. Bison is a program that generates parsers and Flex is a program used to generate scanners. When olsrd is successfully compiled, it is successfully configured by doing the necessary changes in */etc/olsrd/olsrd.conf*. Then the configuration file is copied to */etc/* using the command *cp /etc/olsrd/olsrd.conf /etc/*.

To install OLSRd in ALIX Boards we downloaded the debian package [10] *olsrd_0.6.2-2_i386.deb*. To compile olsrd the used was dpkg –I *olsrd_0.6.2-2_i386.deb*. To configuration procedure is same like did in laptops

C. Clock Synchronization

Network Time Protocol (NTP) is networking protocol for clock synchronization between computer systems over packet-switched, variable latency data networks [12]. Install *ntp* and ntpdate both in server and client systems using *sudo apt-get install ntp* and *sudo apt-get install ntpdate*. Start the ntp service in server system by issuing the command *sudo /etc/init.d/ntp* start or *sudo service ntp restart*. In client system stop the ntp services first and then synchronize to server using ntpdate. When it is synchronized to server time with some offset then start the services of ntp in client. The commands used in client are *sudo /etc/init.d/ntp stop*, *sudo ntpdate server-ip* and *sudo /etc/init.d/ntp start*.

IV. Test bed implementation

When the system design was over nodes are ready with Ad-hoc configuration and with olsrd protocol. We worked with 2 laptops and 8 ALIX boards. The test bed was conducted in the indoor environment of our lab surrounding. For Single-hop Ad-hoc connection no need to run any network protocol. But we have to develop Multi-hop Ad-hoc network. According to the range of *txpower*, we arranged the nodes in outdoor environment and start running olsrd using the command *olsrd –d 2*, where *d* indicates debugging *2* is one of the debug level and it prints the neighbor table and the topology table information. This results the nodes which are of one-hop neighboring and two-hop neighboring through which one-hop neighbor. We tested it up to 7 hops. In this we initially transmitted a text file of different sizes using secure copy [11]. But the objective of this project is to evaluate the performance in packet loss and delay while transferring a video file in multi-hop wireless Ad-hoc test bed along with multi-path.

Testing scripts were written in socket programming using UDP/IP protocol [13] [14] using C language to implement server and client for the communication between them. Here server waits for the video file to receive from client. Before transferring a video file both server and client should be clock synchronized with each other, so that both contains the exact system time. This is implemented using *ntp* and *ntpdate* [12]. The test started with two nodes and ended with 9 nodes. The video file of 9.3 MB (93, 40, 829 bytes) is taken and is split to 570 packets, each of 16, 396 bytes and last packet is of 1961 bytes. Here the data rate was taken as 16*1024 bytes/sec. This video file is transferred for different hops with single and multi flows. To analyze the best performance for the same scenario the file was transferred for minimum of 3 times and finally the average result was considered. The test was conducted for

different hops, different flows and even when the network links are busy.

V. Results

The video file is transferred for different hops with single and many flows. After the completion of all scenario test bed the results were analyzed. In our observation as the number of hops is increasing loss and delay are also increasing. Figure 2 shows the result of loss and delay for different hops. For a particular hop as the number of flows is increasing to reach the same destination the loss is decreasing but the delay is increasing. This scenario is clearly showed in Figure 3 for various hops. When a single node is contributing for different destinations, which are destined in same hop distance but when the destinations are increasing at the same time then loss is increasing and delay is varying. All these results were taken when the network is good i.e. when all the links were good. Though the test bed is static, network links brakes. With the disadvantage of OLSR takes a lot of time to reconstruct the broken link [15]. When only one node is contributing for different flows, even though the links are free and good the received file is not error free.
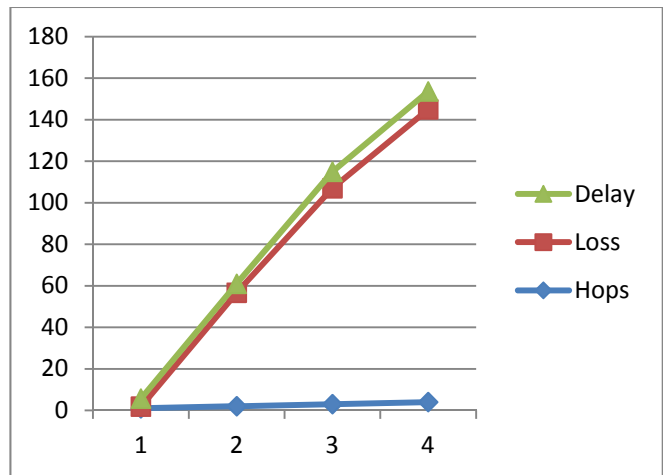
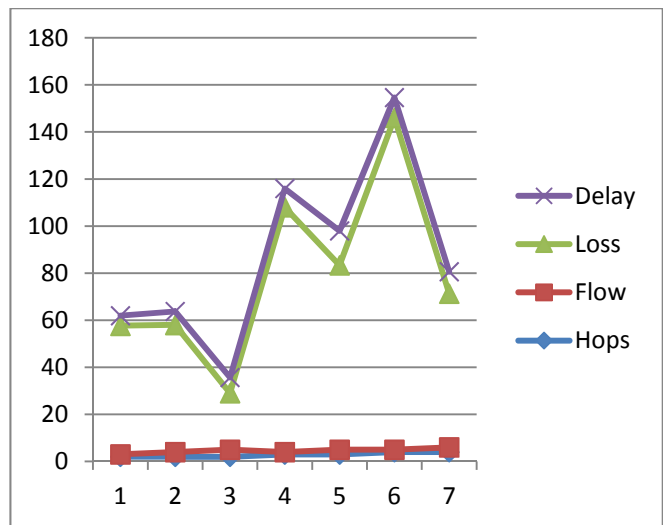Figure 2. Packet Loss and Delay comparison for different hops.

Figure 3. Packet Loss and Delay comparison for different hops as the flows increasing.

VI. Conclusions and Future Work

In this paper we have investigated the performance of a multi-hop Ad-hoc wireless network using OLSR protocol in an indoor environment using real test bed with a number of ALIX boards and 2 laptops. We transferred a video for different hops in different scenarios. Though the network link is good and free, the received file is not error free. However the network topology is stable link quality is not stable and links are breaking. If a single contributes for multiple destinations then loss and delay are more. We are started to work on this trying for better performance over the same protocol, and also plan to do the same using different proposed protocols and have to compare the results.

References

[1] IETF, "Mobile Ad-hoc Networks (manet)", http://www.ietf.org/dyn/wg/charter/manet-charter.html

[2] S. Das, C. Perkins, and E. Royer. Ad Hoc On Demand Distance vector (AODV) Routing. In *Internet Draft, draft-ietf-manet-aodv-11.txt,* work in progress, 2002.

[3] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimal Link State Routing Protocol for Ad Hoc Networks, IEEE INMIC Pakistan. 2001.

[4] D. Johnson, D. Maltz, and J. Jetcheva. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. In *Internet Draft, draft-ietf-manet-dsr-07.txt*, work in progress, 2002.

[5] B. Bellur, R.G. OGIER, and F.L Templin. Topology broadcast based on reverse-path forwarding routing protocol(rtbpf). In *Internet Draft, draft-ietf-manet-tbrpf-06.txt,* work in progress, 2003.

[6] "Voyage Linux" [Online], Available: hhtp://linux.voyage.hk/download.

[7] "Olsr daemon" [Online], Available: http://www.olsr.org/

[8] " Ad hoc Connection" [Online], Available: https://help.ubuntu.com/community/WifiDocs/Adhoc

[9] "OLSR for Linux" [Online], Available: http://tldp.org/HOWTO/OLSR-IPv6-HOWTO/olsrlinux.html

[10] "Debian – Details of package olsrd in sid - packages – Debian" [Online] Available: http://packages.debian.org/unstable/net/olsrd

[11] "Secure Copy" [Online], Available: http://www.computerhope.com/unix/scp.htm

[12] "Network Time Protocol" [Online], Available: http://en.wikipedia.org/wiki/Network_Time_Protocol

[13] "Writing a simple UDP client/server in a Unix environment" [Online], Available: http://www.abc.se/~m6695/udp.html

[14] "Socket Programming Tutorial" [Online], Available: http://www.prasannatech.net/2008/07/socket-programming - tutorial. html

[15] " Comparison Between AODV and OLSR" [Online], Available: https://docs.google.com/viewer?a=v&pid=gmail&attid=0.1&thid=137c030476b49653&mt=application/vnd.ms-powerpoint&url=https://mail.google.com/mail/u/0/?ui%3D2%26ik%3Dabea154493%26view%3Datt%26th%3D137c030476b49653%26attid%3D0.1%26disp%3Dsafe%26realattid%3Df_h33xq1xk0%26zw&sig=AHIEtbQ2VZyFlDxhYyirDF3CiLXJNskD7g&pli=1

[16] Jiazi YI, Sylvain DAVID, Asmaa ADNANE, Benoit PARREIN, Universe de Nantes, Nantes Atlantique Universites, Xavier LECOURTIER Keosys, 1 impasse Augustin Fresnel " Multipath OLSR: Simulation and Testbed"