

Evaluation of OLSR Protocol Implementations using Analytical Hierarchical Process (AHP)

Ashfaq Ahmad Malik and Tariq Mairaj Rasool Khan
PN Engineering College (PNEC)
National University of Science and Technology (NUST)
Karachi, Pakistan

Athar Mahboob
Khwaja Fareed University of
Engineering and Information Technology (KFUEIT)
Rahim Yar Khan, Pakistan

Abstract—Adhoc networks are part of IEEE 802.11 Wireless LAN Standard also called Independent Basic Service Set (IBSS) and work as Peer to Peer network by default. These work without the requirement of an Infrastructure (such as an Access Point) and demands specific routing requirements to work as a multi-hop network. There are various Adhoc network routing protocols which are categorized as Proactive, Reactive and Hybrid. OLSR (a proactive routing protocol) is one of widely used routing protocols in adhoc networks. In this paper an empirical study and analysis of the various OLSR implementations (by different research groups and individuals) has been conducted in light of Relative Opinion Scores (ROS) and Analytical Hierarchical Process (AHP) Online System software. Based on quantitative comparison of results, it is concluded that OLSRd project is most updated and best amongst six variants of OLSR protocol implementations.

Keywords—OLSR; MANET; AHP; Routing Protocols

I. INTRODUCTION AND BACKGROUND INFORMATION

We are working on Mobile Adhoc Networks (MANETs) based on 802.11 WLAN Standard based mobile devices to build a trustworthy collaborative system. Due to peculiar nature of adhoc networks (as compared to infrastructure networks); the normal routing protocols used in infrastructure or Access Point (AP) based network can not be directly applied or used in adhoc networks. There are many routing protocols defined and used as an outcome of research in adhoc networks such as Adhoc On-demand Distance Vector (AODV), Fisheye State Routing (FSR) Protocol, Destination-Sequenced Distance-Vector (DSDV), Optimized Link State Routing (OLSR) etc. These are broadly categorized as Pro-active and Re-active routing protocols depending upon the type of algorithm used. OLSR is one of the commonly and widely used routing protocol in adhoc networks. It helps in multi-hop communication between the peer to peer nodes connected in adhoc network.

A. OLSR Protocol

OLSR is a proactive routing protocol used by MANETs. RFC 3626 [1] was implemented as OLSR daemon (olsrd) in 2004 [2]. As per RFC 3626, the key concept is Multi-Point Relays (MPRs). Unlike link state routing, where every node transmits broadcast messages; in OLSR only MPRs transmit broadcast messages. Only MPRs generate link state information, hence reducing the number of control messages flooding the network. Thirdly, a MPR may choose to report links between itself and MPR selectors. Learning from experiences

of OLSR version 1, RFC 7181 has been issued for latest version i.e. OLSRv2 [3]. It is updated by RFC 7183, 7187, 7188, and 7466.

OLSRv2 is also a table driven, proactive routing protocol which retains the basic mechanisms and algorithms of its predecessor, however, few enhancements have been done in calculation of shortest routes through use of link metric (other than hop count), simplification of messages exchanged and efficient/flexible signaling framework. It is also the further optimizing of classic link state routing protocol and works on concept of MPRs. There are 02 sets of MPRs selected by each router i.e. 'Flooding MPRs' and 'Routing MPRs' used for reduction of flooding and topology, respectively.

1) *Implementations of OLSR*: Various organizations have carried out implementations and research work on OLSR and is summarized as under. These are primarily based on OLSRv1:

- pyOLSR by INRIA [4]. Initially, pyOLSR was implemented by INRIA as a prototype based on RFC 3626, however, now it is obsolete.
- OOLSR by Hipercom-INRIA[5]. Complete re-implementation in C++ based on RFC 3626 by a Hipercom team at INRIA. It supports Windows and Linux.
- SMOLSR-MOLSR by Hipercom-INRIA[6]. Multicast OLSR (MOLSR) and Simple Multicast OLSR (SMOLSR) are extensions of OOLSR. Their function is to get the control and routing information. The data is forwarded through Multicast Data Forwarding Protocol(MDFP) daemon.
- NOA-OLSR by Niigata University in collaboration with INRIA[7]. No Overhead Auto-configuration OLSR (NOA-OLSR) has been implemented in collaboration with INRIA, based on an Algorithm developed by a professor of Niigata University and INRIA.
- NRL OLSR [8]. It is an implementation based on RFC 3626 with some additional features by Naval Research Laboratories. The code is based on INRIA's OLSR Version 3 protocol draft for the IETF [9]. It has Windows and Linux based distributions.
- QOLSR by LRI [10]. It is a Quality of Service (QoS) extension to OLSR for Linux by Laboratoire de

Recherche en Informatique (LRI), France. Qolyester is RFC 3626 complaint and implemented in C++.

- OLSR by GRC [11]. A Networking Research Group (Grupo de Redes de Computadores (GRC)) at Universitat Politcnica de Valncia (UPV) has ported the OLSR implemented by INRIA to Windows 2000 and Pocket PC.
- OLSR by Unik University [2]. It is the most widely used OLSR implementation and is generally called OLSR Daemon (OLSRd). The olsrd works on Windows (XP and Vista, Windows 7), Linux (i386, arm, alpha, mips, xscale), OS X (powerpc, intel, xscale, iPhone), VxWorks, NetBSD, FreeBSD, OpenBSD, Nokia N900, Google phone (Android) etc [12].

2) *OLSRd Plugins – Additional Features:* Plugin is an add-on pluggable program fragment enhancing functionality of some program or system. Plugins in context of OLSRd is the supportability to load dynamically loadable library (DLL) for the purpose of performing different functions and to generate or process private package types. In Linux DLL functionality is available in .so files and in Windows as .DLL file extension. The olsrd plugins design has been chosen due to following reasons [2]:

- To add any custom functionality or package, the source code of olsrd is not required to be changed.
- The plugins can be licensed separately as per conditions of user.
- Any language can be used to code the plugins and can be compiled as dynamic library.
- The plugins have backward compatibility.

Various types of plugins for OLSRd are Tiny Web Server (httpinfo), OLSR Node Information Display (txtinfo), Basic Multicast Forwarding (bmf), Securing OLSR Route (secure), Outputs in Dot Format over TCP Socket (dot-draw), Dynamic Announcement of Uplinks (dyn-gw Announcing), DNS Servers and host names (nameservice), import of external routes from qaugga (qaugga), Performance Graph (pgraph), Distributing P2P Discovery Messages (p2pd), Minimal Example (mini), Tiny Application Server (tas), Detecting OLSRd Freezing (watchdog), optimizing kernel ARP cache from OLSR UDP sniffing (arprefresh), Muticast DNS (mdnsp), Position Update (PUD) etc.

B. Analytical Hierarchical Process (AHP)

AHP is a type of multi-criteria assessment (MCA) technique for analyzing complex decisions. It measures intangibles in relative terms. AHP is a mathematical as well as psychological approach and a structured technique to carry out complex decisions specially applied in group decision making. It was initially studied and researched in the 1970s by Thomas L. Saaty. It has been improved and applied in solving decision problems until now. A good resource on AHP is available at [13]. An AHP process decision making recommends a most suitable choice of alternatives based on user defined criteria. As per this book following steps are involved in an AHP process decision making to recommend a most suitable choice of alternatives based on a defined criteria:

- **Step-1.** The problem is modeled primarily in a hierarchy of three layers as under:
 - The Goal or Main Objective to achieve.
 - The Criteria for evaluation of alternatives.
 - The choice of available alternatives.
- **Step-2.** The elements of hierarchy are pairwise compared based on multiple judgments of each pair of element to establish priorities.
- **Step-3.** The overall priorities of hierarchy are calculated through synthesis of above judgments.
- **Step-4.** Check whether the judgments are consistent and conclude final result based on these judgments.

Various types of tools are available in the market to apply AHP such as BPSMSG AHP Online System (AHP-OS), Priority Estimation Tool (PriEst), AHP Solver, MakeItRational, Open Decision Maker, AHP Analyzer, AHP Software, ABC AHP Decision Making Software, easyAHP, AHP.net etc. We in our research have used AHP-OS.

1) *BPSMSG's AHP-OS:* AHP-OS is web based tool developed by Business Performance Management Singapore (BPSMSG). It is one of the most latest, updated and easy to use web application/tool (developed in php) available online for Multi Criteria Decision Making (MCDM) based on classical AHP by Saaty. It does not cater other MCDM methods such as Fuzzy AHP, Modified AHP(M-AHP) etc and has peculiar advantages, disadvantages and limitations as associated with each method. It calculates weightings or ratio scales (by paired comparison of criterion) and consistency index based on input by the user (either calculated or subjective opinions). Mathematically it is based on calculation of Eigen value problem. The calculation of Eigen value gives the consistency ratio whereas, the dominant normalized right Eigen vector of the matrix gives the scale ratio. It is based on following features which are available to registered users:

- AHP Projects – A hyperlink to handle complete AHP projects including group decision support.
- AHP Priority Calculator – A hyperlink calculate priorities based on pairwise comparisons.
- AHP Hierarchies – A hyperlink for defining complete set of hierarchies, evaluation of priorities and alternatives.
- AHP Group Session – A hyperlink for participating in AHP group sessions.

2) *Application of AHP in Software Selection:* A broad review of application of AHP has been presented in [14], [15]. It includes but not limited to selection, evaluation, benefitcost analysis, allocations, planning and development, priority and ranking, decision-making, forecasting in medicines and related fields. Other areas are personal, social, manufacturing sector, political, engineering, education, industry, government, and others which include sports, management etc.

AHP is also applied to selection of software. Simulation Software [16], Multimedia Authorizing Systems (MAS) [17], Project Management Software [18], ETL Software [19], Data Warehouse System for Large and Small Enterprises in Taiwan

[20], Forecasting Software [21] etc are best examples in this regard. These studies motivated us to apply AHP to different variants of OLSR software to select the best one to suit in our adhoc network based collaborative system project.

3) *Mathematical Modeling in AHP*: A complete and elaborate description of mathematical modeling and application of relevant theorems in AHP is given in [22]. The mathematics used in determining and calculating decision hierarchy and overall result in AHP-OS is given in [23]. The same is summarized in following paragraphs.

- **Scale of Intensity**. The scale of intensity from the 1-9 (denoted by x) is used as an integer for each selection while comparison of paired criteria and alternative. The x is transformed into c (which is used as an element in pairwise comparison matrix) as under:

Linear scale:

$$c = x \quad (1)$$

Logarithmic scale:

$$c = \log_2(x + 1) \quad (2)$$

Root Square scale:

$$c = \sqrt{x} \quad (3)$$

Inverse Linear:

$$c = \frac{9}{(10 - x)} \quad (4)$$

Balanced Scale: When $w = 0.5, 0.55, 0.6, \dots, 0.9$.

$$c = \frac{w}{(1 - w)} \quad (5)$$

Power Scale:

$$c = x^2 \quad (6)$$

Geometric Scale:

$$c = 2^{x-1} \quad (7)$$

- **Row Geometric Mean Method (RGMM)**. RGMM has been used to calculate the priorities P_i , to input the $N \times N$ pairwise comparison of the matrix $A = a_{ij}$. The calculation and normalization is done as under: Calculation:

$$r_i = \exp \left[\frac{1}{N} \sum_{j=1}^N \ln(a_{ij}) \right] = \left(\prod_{j=1}^N a_{ij} \right)^{\frac{1}{N}} \quad (8)$$

Normalization:

$$p_i = \frac{r_i}{\sum_{j=1}^N r_j} \quad (9)$$

- **Consistency Ratio(CR)**. CR is calculated by calculating λ_{max} (the principal eigenvalue) and putting in equation below (calculated by Lonson/ Lamata linear fit):

$$CR = \frac{\lambda_{max} - N}{2.7699N - 4.3513 - N} \quad (10)$$

II. DEFINING OF CRITERIA/ ELEMENTS OF CRITERION

A. General Criteria – Mean Opinion Score (MOS) and Relative Opinion Score (ROS)

A general selection criteria based on ROS (as applied in GeoSharing project [24] for selection of an embedded Operating System) where a relative score from 1-5 has been considered (5 being highest score awarded based on observation or judgment of each observer). This type of scoring is relative to one another (of the projects under consideration) and once carried out by single person can be termed as Relative Opinion Score (ROS). A more relevant ROS can be related as mentioned in Table I. The same type of scoring once conducted through a group of peoples and their average is taken as Mean Opinion Score (MOS).

TABLE I. RELATIVE OPINION SCORE (ROS) AWARDS

Observation	Score
Excellent	5
Very Good	4
Good	3
Satisfactory	2
Just Satisfactory	1
Un-Satisfactory	0

B. Elements of Criterion

The selection a software for practical usage depends on multiple factors. However, following factors are considered vital and have been considered as a selection criteria for selection of OLSR software.

- **Stability of software** – Measure of reliability and robustness that it should not crash and it is usable without bugs/ interruptions.
 - OLSRd and NRL-OLSR are quite stable and we assign it ROS of 4. We have experienced crashing of OLSRd and NRL-OLSR very few times.
 - The QOLSR implementation is tested in OP-NET simulator. Qolyester is a testbed for QOLSR algorithms,hence, may have bugs. As per available documentation certain bugs are already known. We assign ROS of 3.
 - OLSR's prototype implementation/ porting on Windows 2000 and Pocket PC by GRC-UPV may have inherent bugs. Hence, we have awarded ROS of 2.
 - NOA-OLSR is also a Proof of Concept implementation of OLSR with No Overhead and Auto-Configuration by INRIA, France. Its source is not available for installation and testing. Hence, exact stability can not be ascertained. We have awarded ROS of 1.
 - pyOLSR, OOLSR (developed in C++), SMOLSR and MOLSR (extensions of OOLSR) are also implementations by INRIA France. However, the source code is not available on the referred website. Hence, we assign ROS of 0 to all of these variants with respect to ascertaining the stability.

- **Maintainability by Developers** – Are the developers maintaining these distributions through nightly or stable builds on regular basis?
 - OLSRd is regularly being maintained by www.OLSR.org; latest version is OLSRd 0.9.0.3. OLSRv1 is being used for up-dating and development of OLSR V2. We assign ROS of 5.
 - NRL-OLSR is also being maintained by Naval Research Labs USA. Latest distribution is NRL-OLSRd Ver 7.8.1; last updated on 10 Aug 2007. We assign ROS of 2 to it.
 - QOLSR’s most current version is Qolyester-20090302 available online on its website. Since, than it is not updated Hence, we have awarded ROS of 3.
 - NOA-OLSR source is not available for installation and testing. Hence, we have assigned ROS of 0 to it.
 - The source of pyOLSR, OOLSR (developed in C++), SMOLSR and MOLSR (extensions of OOLSR) is also not available and not being updated. Hence, we assigned ROS of 0.
- **Usage** – Is there any developer community which is using the software? User experience of general users and their views are also important factors.
 - As discussed in preceding sub section of Maintainability, same ROS scoring is applied with respect to NRL-OLSR and QOLSR.
 - Considering usage and testing of software by the developer community; we assign ROS of 1 each to NOA-OLSR, OOLSR and PyOLSR.
 - Based on experiences of OLSRd; OLSRv2 is developed and being improved. Moreover, OLSRd has been implemented and used in various adhoc networking projects such as GeoSharing, MANET Manager (SPAN), Byzantium, Commotion, Qual.net etc. Hence, ROS of 5 is awarded for Usage criteria element.
- **Security** – Are security features appropriately addressed in the software and the known vulnerabilities adequately addressed?
 - OLSRd has a security plug-in and we assign ROS of 4.
 - Other variants lacks security feature, hence, we have assigned ROS of 0 to each one of them.
- **Cross-platform** – Does the software support multiple OS platforms such as Windows, Linux, Mac, Android etc.
 - OLSRd is developed for multiple platforms including Windows, Linux, Mac and Android. We award ROS of 5.
 - The QOLSR is developed for Linux platform. We assign ROS of 2.
 - The NRL-OLSR is also supports multi-platforms and we assign it ROS of 4.
 - NOA-OLSR is also implemented for Linux. We award ROS of 1.
 - pyOLSR is also supports multi-platforms i.e. Windows, POSIX and Linux. We assign ROS

of 3.

- OOLSR is also Windows and Linux based. We assigned ROS of 3.

- **Other Features** – Are multiple features (other than the basic design) provided?
 - OLSRd provides multiple features through plugins as discussed in sub-section above. We award ROS of 4.
 - The QOLSR provides QoS feature. We assign ROS of 1.
 - The NRL-OLSR supports fuzzy-sighted routing and Simplified Multi-cast Forwarding. We assigned ROS of 2.
 - NOA-OLSR supports No Overhead Auto-configuration; a very important feature in adhoc environment. We award ROS of 1.
 - pyOLSR works with basic OLSR. We assign ROS of 0.
 - OOLSR is improved to provide SMOLSR and MOLSR. We assigned ROS of 2.

C. Summary of ROS

The summary of ROS as per our opinion based on discussion in sub-section II-B are as given in Table II.

TABLE II. RELATIVE OPINION SCORE (ROS) - VARIANTS OF OLSR

Software/ Criteria	OLSRd	pyOLSR	OOLSR	NOA OLSR	NRL OLSR	Q OLSR
Stability	4	0	0	1	4	3
Maintenance	5	0	0	0	2	3
Usage	5	1	1	1	2	3
Security	4	0	0	0	0	0
Cross-platform	5	3	3	1	4	2
Other Features	4	0	2	1	2	1
Total	27	4	6	4	14	12

III. EXPERIMENTATION WITH BPMSG’S AHP-OS

With this definition of basic criteria we move on to our experimentation with BPMSG’s AHP-OS. Register with AHP-OS website [22] and login with the provided user name and password. Further steps involved are discussed in ensuing paragraphs.

A. Defining Hierarchy

We defined the hierarchy using the basic node as “Selecting OLSR Software” with node leaf or sub-categories as Stability, Maintenance, Usage, Security, Multi-platform support and Other Features etc. The overall OLSR selection AHP hierarchy along-with requisite criteria and alternatives is as shown in Figure 1.

B. Compare Criteria

Each category of criteria is pairwise compared to find that which criterion has more weight or importance. In our opinion Stability, Maintenance and Security has more importance as compared to Usage, Multi-platform and Other Features criterion of OLSR software. Each pair of criterion was compared in light of following AHP Scale:

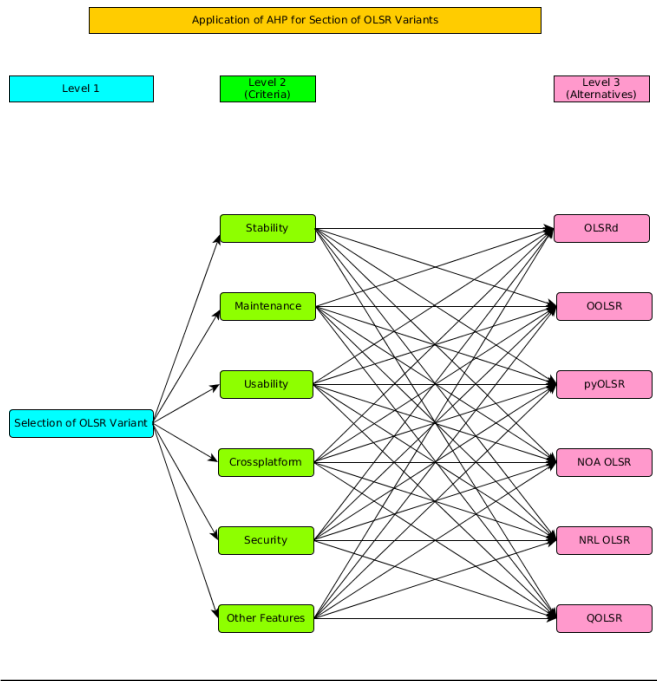


Fig. 1. Overall AHP Hierarchy for Selection of OLSR

Equal Importance as 1, moderate importance as 3, strong importance as 5, very strong importance as 7, Extreme importance as 9, whereas 2, 4, 6, 8 are the values in-between afore stated values.

The selection of pairwise criterion is shown in Figure 2. The overall result of pairwise comparison of each criterion element is shown in Figure 3.

C. Evaluation of Alternatives

On completion of pairwise comparison of criterion, the evaluation of alternatives i.e. OLSR software available in the open source community are also pairwise compared to one another based on subjective opinions or actual measurements. The software also provides group input based decisions as well. The pairwise comparison is made based on the ROS as given in Table II. As an example the pairwise comparison of only one criterion for all the alternatives is shown in Figure 4. Overall status of alternatives is shown in Figure 5. The threadbare analysis of the results recorded in ROS Table II and AHP-OS's Figure 5 reveal that AHP has more granularity with respect to analysis of each criteria element as compared to ROS.

D. Summary of Results

Six OLSR alternatives have been compared in light six criterion elements. Each criterion weighting based on the applicable importance is summarized in graph at Figure 6. Similarly, different variants of OLSR and their overall percentages are summarized as shown in Figure 5. We can clearly see that OLSRd is ranked first, followed by NRL-OLSR as second and QOLSR as third.

A - wrt Selecting OLSR Software - or B?		Equal	How much more?							
1	Stability or Maintenance	1	2	3	4	5	6	7	8	9
2	Stability or Usage	1	2	3	4	5	6	7	8	9
3	Stability or Security	1	2	3	4	5	6	7	8	9
4	Stability or Cross-platform	1	2	3	4	5	6	7	8	9
5	Stability or Other Features	1	2	3	4	5	6	7	8	9
6	Maintenance or Usage	1	2	3	4	5	6	7	8	9
7	Maintenance or Security	1	2	3	4	5	6	7	8	9
8	Maintenance or Cross-platform	1	2	3	4	5	6	7	8	9
9	Maintenance or Other Features	1	2	3	4	5	6	7	8	9
10	Usage or Security	1	2	3	4	5	6	7	8	9
11	Usage or Cross-platform	1	2	3	4	5	6	7	8	9
12	Usage or Other Features	1	2	3	4	5	6	7	8	9
13	Security or Cross-platform	1	2	3	4	5	6	7	8	9
14	Security or Other Features	1	2	3	4	5	6	7	8	9
15	Cross-platform or Other Features	1	2	3	4	5	6	7	8	9
CR = 7.3% OK										

Fig. 2. Pairwise Comparison of Criterion Elements

Decision Hierarchy		
Level 0	Level 1	Global Priorities
Selecting OLSR Software	Stability 0.2373	23.7 %
	Maintenance 0.2373	23.7 %
	Usage 0.1548	15.5 %
	Security 0.2373	23.7 %
	Cross-platform 0.0666	6.7 %
	Other Features 0.0666	6.7 %
OK. Submit for group eval or alternative eval. Alternatives		1.0

Fig. 3. Pairwise Comparison Results – Criterion Elements

IV. CONCLUSION AND FUTURE WORK

OLSRd (by OLSR.org) has been finalized as a better project and a choice for practical implementation for use in a collaborative system being designed in an adhoc networking environment. The empirical study based on quantitative comparison of ROS and AHP criteria elements; ranks OLSRd as First and NRL-OLSR as Second choice for implementation in practical adhoc networks. OLSR2 based on OLSRv2 is considered as the latest implementation based on OLSRd by OLSR.org and can be a better choice for future projects

A - wrt Stability - or B?		Equal	How much more?							
1	<input checked="" type="radio"/> OLSRd or <input type="radio"/> PyOLSR	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input checked="" type="radio"/> 8	<input type="radio"/> 9
2	<input checked="" type="radio"/> OLSRd or <input type="radio"/> OOLSR	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input checked="" type="radio"/> 8	<input type="radio"/> 9
3	<input checked="" type="radio"/> OLSRd or <input type="radio"/> NOA-OLSR	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input checked="" type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
4	<input checked="" type="radio"/> OLSRd or <input type="radio"/> NRL-OLSR	<input checked="" type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
5	<input checked="" type="radio"/> OLSRd or <input type="radio"/> QOLSR	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input checked="" type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
6	<input checked="" type="radio"/> PyOLSR or <input type="radio"/> OOLSR	<input checked="" type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
7	<input type="radio"/> PyOLSR or <input checked="" type="radio"/> NOA-OLSR	<input type="radio"/> 1	<input checked="" type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
8	<input type="radio"/> PyOLSR or <input checked="" type="radio"/> NRL-OLSR	<input type="radio"/> 1	<input type="radio"/> 2	<input checked="" type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
9	<input type="radio"/> PyOLSR or <input checked="" type="radio"/> QOLSR	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input checked="" type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
10	<input type="radio"/> OOLSR or <input checked="" type="radio"/> NOA-OLSR	<input type="radio"/> 1	<input type="radio"/> 2	<input checked="" type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
11	<input type="radio"/> OOLSR or <input checked="" type="radio"/> NRL-OLSR	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input checked="" type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
12	<input type="radio"/> OOLSR or <input checked="" type="radio"/> QOLSR	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input checked="" type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
13	<input type="radio"/> NOA-OLSR or <input checked="" type="radio"/> NRL-OLSR	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input checked="" type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
14	<input type="radio"/> NOA-OLSR or <input checked="" type="radio"/> QOLSR	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input checked="" type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
15	<input checked="" type="radio"/> NRL-OLSR or <input type="radio"/> QOLSR	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input checked="" type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9

CR = 0% Please start pairwise comparison

Calculate Result AHP Balanced scale

Fig. 4. Pairwise Comparison of Alternatives–Stability

after testing in practical scenarios. Application of AHP-OS decision making helps and provides the objective mathematics to process the inescapable subjective and personal preferences of an individual or a group in making a decision based on various criterion. We have applied and configured a level-2 hierarchy. We intend to apply more relevant sub-criterion to a group of hierarchy such as selection of most suitable adhoc networking project for building a nomadic collaborative information system.

ACKNOWLEDGMENT

We acknowledge the support of NUST-PNEC for carrying out this research by providing necessary funding for procurement of COTS devices. We also thank BPSMG for the free access to on-line AHP-OS Software.

REFERENCES

[1] Network Working Group, "Optimized Link State Routing Protocol (OLSR)." RFC 3626, October 2003.

[2] A. Tnnesen, "Impementing and extending the Optimized Link State Routing Protocol." Master's thesis, Department of Informatics, University of Oslo, August 2004.

[3] Internet Engineering Task Force (IETF), "Optimized Link State Routing Protocol (OLSR) Version 2 (RFC7181)." RFC 7181, April 2014.

[4] Hipercom-INRIA, "pyOLSR." Online Web Page, July 2003. Accessed on 01 August 2015.

[5] Hipercom-INRIA, "OOLSR." Online Web Page, November 2004. Accessed on 01 August 2015.

[6] Hipercom-INRIA, "SMOLSR-MOLSR." Online Web Page, August 2005. Accessed on 01 August 2015.

[7] Hipercom-INRIA and Niigata University, "NOA-OLSR." Online Web Page, 2005. Accessed on 01 August 2015.

[8] Naval Research Laboratories (NRL), "The NRL OLSR Routing Protocol Implementation." Online.

[9] Ron Lee and Joe Macker, "OLSRD HOWTO, Version 1.0." Online, October 2005.

[10] Laboratoire de Recherche en Informatique (LRI), France, "QOLSR." Online Web Page, 2005. Accessed on 01 August 2015.

[11] Grupo de Redes de Computadores (GRC), Networking Research Group-Universitat Politcnica de Valncia (UPV), "OLSR protocol implementation." Online, July 2009. Accessed on 01 August 2015.

[12] Andreas Tnnesen and Thomas Lopatic and Hannes Gredler and Bernd Petrovitsch and Aaron Kaplan and Sven-Ola Tcke, "olsrd - an adhoc wireless mesh routing protocol ." Online Web Page, 2008 (updated). Accessed on 01 August 2015.

[13] T. L. Saaty, *Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World*. RWS Publications, 2008.

[14] O. S. Vaidya and S. Kumar, "Analytic hierarchy process: An overview of applications," *Elsevier's European Journal of Operational Research*, pp. 1–29, April 2004.

[15] T. L. Saaty and L. G. Vargas, *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process; International Series in Operations Research & Management Science*, ch. Chapter 2: The Seven Pillars of the Analytic Hierarchy Process, pp. 23–40. Springer Science+Business Media NY, 2012.

[16] S. Erees, E. Kuruolu, and N. Morali, "An application of analytical hierarchy process for simulation software selection in education area," *Frontiers in Science*, vol. 3, pp. 66–70, March 2013.

[17] Vincent S. Lai and Robert P. Trueblood and Bo K. Wong, "Software Selection: A case study of the application of the Analytical Hierarchical Process(AHP) to the selection of a Multimedia Authoring System (MAS)," *Elsevier's Information and Management*, vol. 36, pp. 221–232, January 1999.

[18] B. Kutlu, A. Bozanta, E. Ates, S. Erdogan, O. Gokay, and N. Kan, "Project Management Software Selection Using Analytic Hierarchy Process Method," *International Journal of Applied Science and Technology*, vol. 4, pp. 113–119, November 2014.

[19] M. Hanine, O. Boutkhoul, A. Tikniouine, and T. Agouti, "Application of an integrated multi-criteria decision making AHP-TOPSIS methodology for ETL software selection," *SpringerPlus*, 2016.

[20] H.-Y. Lin and P.-Y. Hsu, "Application of the Analytic Hierarchy Process on Data Warehouse System Selection Decisions for Small and Large Enterprises in Taiwan," *International Journal of The Computer, the Internet and Management*, vol. 15, pp. 73–93, December 2007.

[21] A. Pekin, G. Ozkan, O. Eski, U. Karaarslan, G. Ertek, and K. Kilic, "Application of the Analytic Hierarchy Process (AHP) for Selection of Forecasting Software," *5th International Symposium on Intelligent Manufacturing Systems, Sakarya, Turkey*, 2006.

[22] K. D. Goepel, "BPMSGs AHP Online System." Online, May 2014. Accessed on 02 Sep 15.

[23] K. D. Goepel, "BPMSG AHP Excel Template with multiple Inputs." Online, December 2013.

[24] L. Lamouline and V. Nuttin, "The GeoSharing project: An Openmoko geoposition sharing system," Master's thesis, École Polytechnique de Louvain, Université catholique de Louvain, 2011.

Decision Hierarchy								
Level 0	Level 1	Global Priorities	OLSRd	PyOLSR	OOLSR	NOA-OLSR	NRL-OLSR	QOLSR
Selecting OLSR Software	Stability 0.2373	23.7%	0.0966	0.0081	0.0078	0.014	0.0748	0.036
	Maintenance 0.2373	23.7%	0.1339	0.0099	0.0103	0.0103	0.0276	0.0453
	Usage 0.1548	15.5%	0.0897	0.0086	0.0086	0.0086	0.0151	0.0241
	Security 0.2373	23.7%	0.1384	0.0198	0.0198	0.0198	0.0198	0.0198
	Cross-platform 0.0666	6.7%	0.0239	0.0089	0.0089	0.0043	0.0164	0.0043
	Other Features 0.0666	6.7%	0.036	0.003	0.0089	0.0049	0.0089	0.0049
OK. Submit for group eval or alternative eval. Alternatives		1.0	51.8%	5.8%	6.4%	6.2%	16.3%	13.4%

Fig. 5. Overall Result of Pairwise Compared Alternatives and Criterion Elements

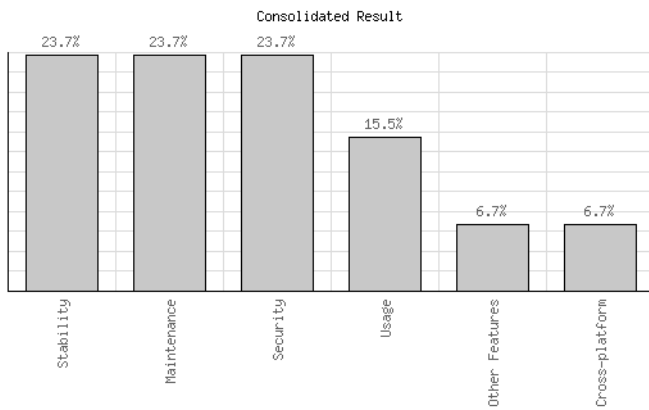


Fig. 6. Summary of Overall Percentage Weight of Criterion Elements

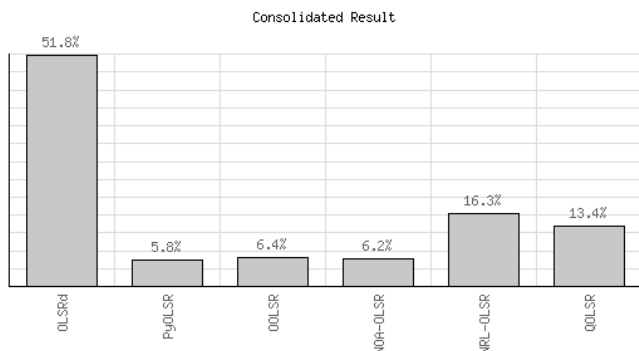


Fig. 7. OLSR Variants Percentages – Participant 1

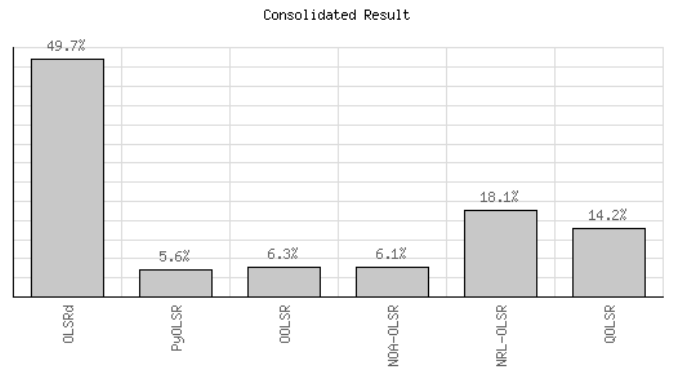


Fig. 8. OLSR Variants Percentages – Participant 2

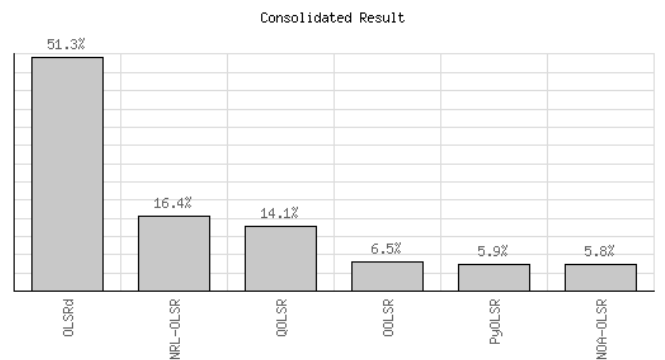


Fig. 9. Overall OLSR Variants Percentages - Average of Both Participants