Design and Implementation of ID based MANET Auto-configuration Protocol

S. Khalid[1], A. Mahboob[2]

[1] Hamdard University Karachi, Pakistan
[2] Dean Electrical DHA Suffa University Karachi, Pakistan
shahrukh_khalid@hotmail.com, atharmahboob@yahoo.com

Abstract: Auto-configuration protocols are used for assignment of unique IP addresses to nodes in Mobile ad hoc networks. Without the assignment of unique IP addresses, service provisioning between the nodes is not possible. Such protocols use various heuristics to ensure the uniqueness in IP address assignment; such aspects increase the overall complexity in MANET system design. Moreover the overriding role of IP address as an ID in Application layer and Locator in routing space is a bottleneck in future wireless network (FWN) design. Contemporary FWN research is focusing on ID/Locator split concept designs. In this paper we propose an ID/Locator based architecture for MANETs which also solves auto-configuration requirements for MANETs. Our proposed architecture is an adaptation from available ID/Locator split concepts for infrastructure oriented networks for usage in MANET context. The designed protocol uses identifiers for node identification, node discovery and traffic flow between end points. The protocol support provision for running contemporary IP oriented services. We have also verified various use cases of our proposed protocol through Linux based implementation.

Keywords: Auto-configuration, Mobile adhoc networks, Robust header compression (ROHC), Private address map, MANET system design, TCP/IP stack, Linux

1. Introduction

Mobile Ad hoc networks are a special category of networks exclusively classified on the basis of attributes like infrastructure-less-ness and Dynamic topology. Due to the dynamic nature and inherent infrastructure-less architecture, solutions developed for configuration and deployment of infrastructure oriented networks cannot be directly applied in MANETs. Present Internet architecture is dependent upon the usage of IP address as a host identifier in the Application Layer and as well as a locator in the routing space. Without the assignment of unique IP addresses to a host, provisioning of services between hosts is not possible. The uniqueness of IP addresses is well settled in infrastructure oriented networks; however, Auto-configuration in the context of MANET is not a trivial problem due to inherent dynamic characteristics of Mobile ad hoc networks.

Infrastructure based IP address assignment solutions do not suffice the need for auto-configuration. Dynamic Host Configuration Protocol (DHCP) [1] and its modified form for IPV6 addressing DHCPV6 [2], rely on the use of centralized servers for ensuring unique IP address assignment. Stateless auto-configuration mechanism for IPV6 [3], initially builds a link local address and sends this address using neighbor discovery protocol (NDP) to its one hop peers [4]. As relying on a single DHCP server for IP Address configuration in MANETs will cause a single point of failure in dynamic MANET topology whereas such a server implementation will require multi-hop communication to reach and function which is different from infrastructure based network configuration and not supported in the MANET context. Moreover, stateless IPV6 auto-configuration is also based on exchanging messages by one hop peers and multi-hop communication is not supported inherently. In order to ensure the assignment of unique IP addresses in MANETs rigorous studies have been conducted for design of Auto-configuration protocols. These protocols are based on varied nature of heuristics. L.Villalba [5], N. Wangi et.al [6] and H. Zhou et.al [7] have conducted rigorous reviews in the field of Address auto-configuration and elaborated large number of such protocols. In order to emphasize the need of such mechanisms consider Figure 1. which illustrates dynamic behavior and peculiar requirement of address assignment in MANET. Consider the blue network of Figure 1. in which unique addresses are required to be assigned. As all nodes are not at one hop distance so regular infrastructure based protocols will not work. Suppose if some protocol is able to assign unique addresses to the nodes in question then how the situation will be managed if some nodes leave the MANET. Can the already configured addresses to the nodes be assigned to any other incoming nodes? Moreover, how the mechanism will handle the reappearance of nodes.

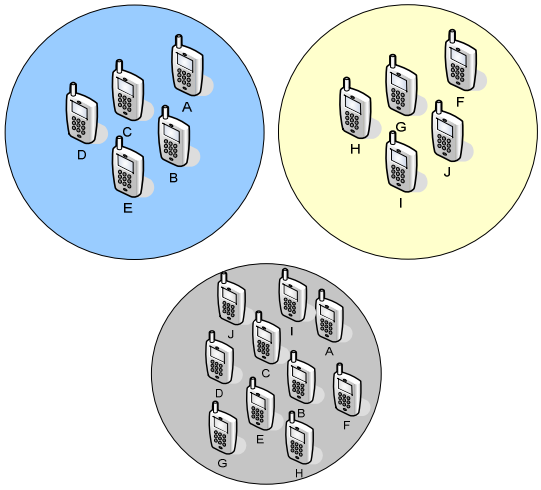

Figure 1. MANETs dynamic topology

Now consider two MANET clouds in blue and yellow illustrated in Figure 1, in which nodes have been assigned distinct IP addresses through some mechanism. If a node in

the right network moves to the left and tries to become the part of the network then how the mechanism could ensure the uniqueness of IP addresses in case there exists duplicate addresses in the newly joined network. Although the protocols can handle the situations of MANET merging, partition and assignment of addresses to a newly formed MANET but overall complexity of real MANET system design is considerably increased. Moreover, additional control overhead is added. After the assignment of IP address through some underlying auto-configuration mechanism the overriding role of IP address Figure 2. is also a bottleneck in the MANET system design [8]. Therefore, in order to address issues of auto-configuration as well as overriding role of IP address we have designed an ID based auto-configuration protocol. The real motivation behind this work is not only to design the protocol but also to implement and know the practical realization and design challenges in MANET context. Practical realization of our design is based on the usage of WiFi enabled devices [9-10]. Section 2 provides details of recent Auto-configuration protocols. Moreover some protocols and architectures which are based on ID/Locator Split concept are discussed. Section 3 gives the practical considerations of our adaptation of such concepts. Section 4 is description of protocol design and implementation details. Section 5 gives details of our designed ID based routing mechanism used for end to end path discovery and IP Application process flow. Section 6 gives the experimentation details and results and Section 7 gives a Qualitative comparison of proposed protocol with present auto-configuration protocols and Section 8 concludes the paper.

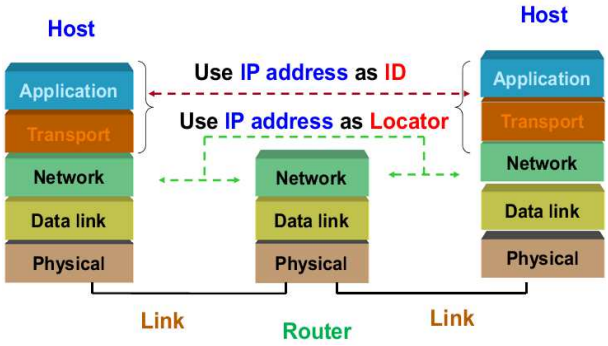

Figure 2. Standard Internet Protocol Based routing

2. Related Work

2.1 Auto-configuration Protocols & Architectures

There are numerous Auto-configuration protocols for MANET IP address configuration. Jose Cano Reyes et al. [11-12] proposed a blue tooth based Auto-configuration architecture for MANETs known as Easy MANET. For configuration of unique IP address the architecture relies on the use of a blue tooth server. The architecture also binds IP address with a user photo and it is referred as Visual DNS system. The discovery mechanism works by exchanging UDP messages. The content of the message contains list of available MANET members. In case a node finds a new member the details are downloaded using TCP connection. The dependency of the Bluetooth server for configuration tasks means if there is non-availability of Bluetooth server then node configuration is not possible. L. Villalba et al. [13] proposed Distributed dynamic host configuration protocol (D2HCP). Each node possesses disjoint set of IP addresses. In case a new node wants to join the MANET then it initiates MAC based communication with the nodes available in its radio range. In case the neighbor node has free available addresses it assigns it to the new node. In case of non availability of addresses a network wide broadcast for IP address is conducted. Upon reception of available IP addresses the new node is assigned an IP address. The protocol uses OLSR for node synchronization and duplicate address detection. Same author has proposed an extension to D2HCP [14] with network merging support. M. Al-Mistarihi et al. [15] proposed a tree based topology oriented auto-configuration mechanism. Each node can have any of three roles which are root node, leader node or normal node. Root node is the main node in the protocol which maintains the records of the leader nodes and their address information in its database and performs tasks of network partitioning and merging. Leader nodes possess disjoint set of IP addresses for assignment of IP addresses to incoming nodes. Normal nodes do not have any special function except used for routing in case of non-availability of leader in a particular area. U. Ghosh et.al [16-17] proposed scheme named as ID based IP configuration scheme (IDDIP). The scheme is based on a one way hash function installed at each node prior deployment of MANET. Each node is capable of assigning IP address to the incoming node and act as a proxy node. Proxy node authenticates the incoming node using public key cryptography; similarly the incoming node also authenticates the sanctity of the proxy node in the same manner. Along with a unique IP address, each node is identified by a unique ID tuple i.e. <*node ID, IP address*>. Z. Slimane [18] proposed security extension to IPV6 Auto-configuration protocol for MANETs. A. Abdelmalek et.al [19] proposed security extension to MANETconf protocol.

2.2 ID/Locator Based Protocols & Architectures

The overriding role of IP address as an ID as well as identifier is not only the problem of MANET but the infrastructure based networks are also facing problem due to this peculiar attribute and it is a bottleneck in the design of Future Wireless Networks (FWN). Leading network researchers have provided critique on the overriding role of IP address [8]. A number of architectures based on the separation or disassociation of the Identifier and Locator of IP address have been proposed and a new field of ID/Locator Split Concept has emerged. The basis of such architectures is the seminal paper of I. Stoica et.al [20] in which internet Indirection Infrastructure (I^3) was presented.

The basic architecture of I^3 is based on rendezvous based communication. Merriam Webster [21] defines rendezvous as:

"A place appointed for assembling or meeting"

This model and the recent models which follow I^3 architecture are based on the introduction of an infrastructure which performs the role of indirection. A receiver can place a trigger, which is a pair of its address and its identifier (id,R) into I^3 infrastructure. The sender can send the data to the receiver based upon pair of (id,data). When the data is received by the I^3 it provides an indirection and sends the

data to the receiver. Figure 3. and Figure 4. illustrate the indirection and I^3 model respectively.

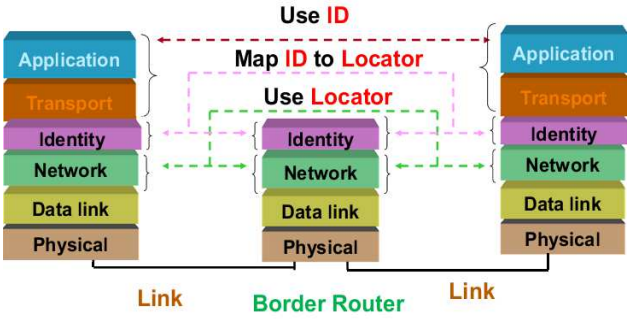

Figure 3. ID Locator Based Routing

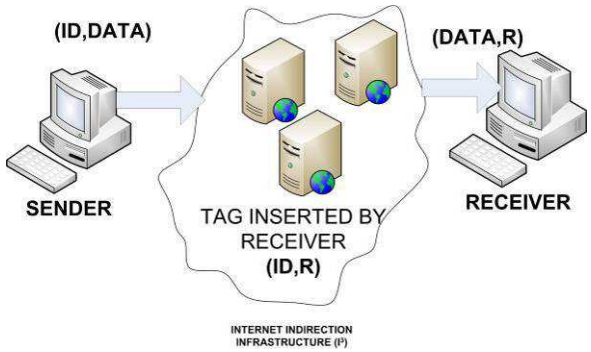

Figure 4. High Level abstraction of ID/Locator Split Concept

MILSA [22-23] Mobility and Multi-homing Identifier Locator Split Architecture proposes the use of composite form of Identifiers. An identifier as per MILSA is composed of a flat part and one hierarchical part. The flat part uniquely defines an object within a particular domain. The hierarchical part defines the unique domain for which a user is a part. Hierarchical border gateway routers are used for mapping between Identifiers and locators. These routers are referred as Realm Zone Bridging servers (RZBS). Enhanced MILSA [24] architecture proposes the usage of a composite ID as a combination of human readable and cryptographic ID as a single node identifier. C. So-In et.al. [25] proposed virtualization architecture based on ID/Locator split concept. Their proposal introduces the usage of virtual machines to build the indirection infrastructure. HIMALIS (Heterogeneity Inclusion and Mobility Adaption through Locator ID Separation in New Generation Network) [26] approaches the ID/Locator split concept by introducing an extra layer between network and transport layer. This layer is the identity sub layer. The function of this upper layer is to support the session identification for the upper layers. Security features for the proposed infrastructure are based on asymmetric keys. Y. Wang et.al [27] proposed an Identifier overlay network (ION) over IP based networks for supporting mobility. P. Martinez-Julia et.al [28] proposed a clean slate approach of ID based architecture design. An idea of domain trusted entity is used in which security and authentication features are based on identities of users. Due to a new architecture current IP based services are required to be modified to run on the proposed architecture. Y. Wang et.al [29] proposed to use IMSI number as identifiers for identity overlay networks.

3. Important Design Considerations and Challenges

In order to design an auto-configurable architecture for Mobile ad hoc networks based on ID/Locator split concept various adaptations and modifications are necessary. ID/Locator split concept is based on the usage of indirection infrastructure in the form of servers, where as in case of Mobile ad hoc networks infrastructure is not available, therefore current ID/Locators architecture cannot be directly applied in MANET context.

Another aspect which is necessary in MANET environment is to provide efficient services. As MANETs are ad hoc networks where there are sparse power resources. If uncompressed data is sent through the nodes then batteries of nodes will drain in a drastic manner. To ensure efficiency, compression schemes for header compressions are required.

There are various problems of adopting header compression schemes in MANET which mainly are dynamic topology, node failure and loss of packet and control information. Moreover, these schemes are IP oriented thus in order to make their usage possible nodes should have unique addresses configured in their interfaces for proper routing of data packets upon change of IP addresses as continuity of compression/decompression cannot be guaranteed [30] – [33]. Moreover, if such a compression scheme is adopted then it will become necessary to compress and decompress the IP Packets at every hop which is an unnecessary processing overhead.

Robust header compression (ROHC) scheme [34] is most popular scheme for compression of IP, ESP, RTP & UDP packets. This protocol maintains a context between compressors and decompresses to recover header information. In dynamic topology of MANETs, if this context information is lost then the headers cannot be recovered. We have introduced the use of Robust Header Compression (ROHC) [34] in our proposed architecture for MANETs context. To the best of our knowledge ROHC is used for point to point links like [35-36] and no design has been proposed for adaptation of ROHC in multi-hop communication between end hosts.

Another necessary aspect is that the design must allow provision of routing or path selection between end points through the available nodes using ID oriented architecture.

4. System Design and Implementation

In order to implement the architecture we have chosen Linux Kernel v 3.2.14 Kernel whereas Ubuntu 12.04 [37] is used as the operation system. The development language

is C++ using GCC version 4.6. Figure 5. gives a high level interaction illustration of networking applications in Linux. Linux process space can be divided into two layers User Space where user applications and other utilities are run. Kernel space is a space where operating system functions like memory management, scheduling, and device drivers like core functions are available. There are high level TCP/UDP sockets applications in user space to interoperate between networking applications e.g. VOIP clients, Chat programs, FTP, HTTP etc. and the Kernel protocol stack. These sockets are used for connectivity between the end points, A. Tudzarov et. al [38] elucidated the end-end socket connectivity of IP based applications. For supporting ID oriented architecture we have developed an integrated system as shown in Figure 8. the details of its subsystem and their interaction is listed in the ensuing paragraphs. We will first explain the routing process flow in Section 5.1, as it is the first step for configuration of various entries in the system. Then we will explain the IP based application process flow in Section 5.2.

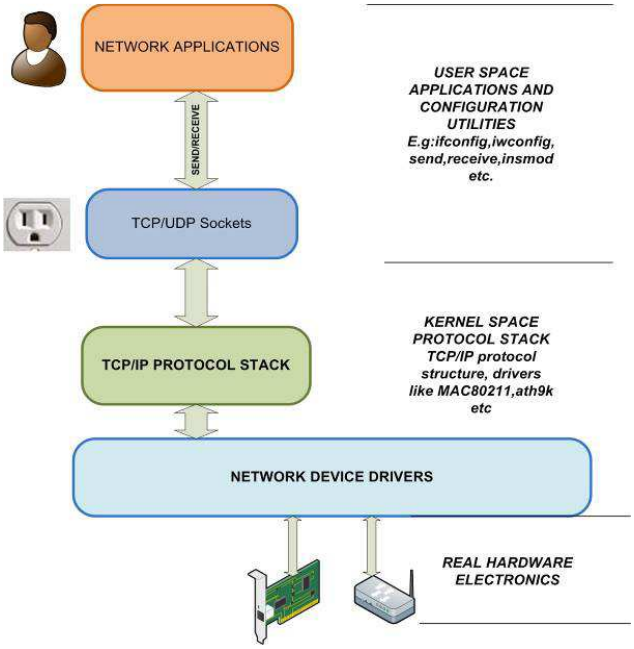

Figure 5. High level abstraction of Linux Networking applications

4.1 Protocol Packet Structure

The Packet structure of the protocol is illustrated in Table I. The protocol uses 04 in No. Identifiers where ID_1 is the actual source node, ID_2 is the actual destination node, ID_3 is the intermediate source node of the routing path and ID_4 is the intermediate destination node of the routing path. All IDs are 64 bit (8 byte) long integers. The protocol field is a 16 bit (2 byte) long integer. The starting bytes are reserved for Radiotap header which is necessary for the Wifi driver [39].

4.2 Sub Systems of ID Based Protocol Architecture

4.2.1 Private Address Map

Linux system gives provision such that a node can configure virtual Ethernet locally and can privately assign IP Addresses to the virtual interfaces. The same provision is used in our design and a single node can configure a number of private IP addresses for other nodes in MANET. A node will instantiate a virtual interface and can assign a unique private address for the node for which it requires communication. As the unique assignment is done privately at the node level therefore no further address detection is required in the whole MANET. Private address MAP also gives provision of running IP oriented applications.

4.2.2 Net Filter System

A net filter system is used for filtering the generated traffic of the IP based application. Before further progression in the protocol stack this system captures the packets and passes it to further modules. The Netfilter system is developed by using two libraries Libnetfilterqueue [40] and Libnfnetlink [41]. These libraries give user space packet handling provision for the packets in Kernel protocol stack. For driving the packets to the filter hooks Linux provides facilities of entering IPTABLES rules [42] through which queues can be assigned to filter any desired IP Packet based on IP address, UDP port number, etc.

4.2.3 Network Address & ID Translation (NAIDT)

Each end host communicates using identifiers. At the start of communication each host configure into the other hosts NAIDT module its Private IP address. A NAIDT system keeps mapping between the Identifier and Private IP addresses at each node using a NAIDT table. This table is a Map type data structure for fast retrieval and insertion. Another function of NAIDT system is to insert the particular Identifiers to the packet being processed.

4.2.4 Compression& Decompression system

This system performs the compression of the generated IP packets before sending it out from the wireless interface and performs decompression of the intended packet required to be processed by the active IP based application. This system is based on ROHC Library [34] Due to the nature of MANET we have used unidirectional mode of ROHC Library.

4.2.5 Packet Socket System

The Packet Socket System is based on the use of Packet Sockets of Linux [43]. When the complete Packet has been constructed by all the subsystems the Packet Socket plays the role of injecting the Packet to Wifi Driver which eventually transmits the Packet to Air. The Packet Socket System is also used for reception of incoming Packets from the Wifi Driver.

Table I. ID Based Protocol Structure

13 Bytes	08 Bytes	08 Bytes	08 Bytes	08 Bytes	02 Bytes	0-2000 Bytes
Radiotap Header	*ID1* (Source)	*ID2* (Destination)	*ID3* (Intermediate Source)	*ID4* (Intermediate Destination)	*Protocol*	*Data*

Upon reception of Packet the Packet Socket System feeds the packet to the other systems for further processing. This system is also used for injection of Packet to the Protocol stack for consumption by the IP Based Application after necessary NAT function and decompression.

4.2.6 Routing Daemon

The first step of communication between the nodes is to find the path or route through the topological formation of Nodes in MANET. Available Routing Protocols are based on IP based identification of nodes therefore they are not suited for our design. We have selected to use Reactive based approach so as to reduce the control traffic. AODV [44] is very popular as a reactive protocol. However due to complexity of data structures we have opted to adapt a simplified version of AODV known as AODVjr. We have modified it structurally and in some interaction mechanism for our design suitability. AODVjr has very good performance as compared to full version AODV as found by I. Chekers et.al [45]. Functioning of AODV can be found in survey of reactive protocols conducted by H. Zafar et. al [46].

5. System Process Flows

5.1 Routing or Path Finding Process Flow

Each node is distinguished by a unique Identifier $\mathbf{I_x}$. I_x can take any type like CNIC Number, IMEI Number, IMSI Number, Vehicle Number, Driving License Number, etc. A node has two in Number tables **NAIDT table** for network address translation based on identifiers and **Routing Table** for ID based Routing. Each node is having a real IP address $\mathbf{IP_x}$ which this node can also use for communicating in an infrastructure based mode like that node can be connected to a fixed network using this real IP address. Each node has a set of virtual or private map of IP addresses **PIP(*x*)** e.g. A's map can have: **PIP(A) = {B(A), C(A), D(A),E(A),…}** obviously **A(A)** does not exist in this map as A already has a real IP Address I_A. Moreover each Node has $\mathbf{CIP_x\ (y)}$ is the configured private address map configured by other nodes at the time routing initiation into each other's NAIDT table. In case of Node A $\mathbf{CIP_A(y) = \{CIP_A(B), CIP_A(C), CIP_A(D),.\ \}}$ does not contain any configured IP address $CIP_A(A)$, as A is the node itself. Let $\boldsymbol{I_A}$ **= 10.129.5.1** be A's real address, A's NAIDT table looks as shown in Figure 6. where I_x is the primary key, we do not have $I_x=I_A$ as this the node itself. When Source Node A wants to communicate with Destination Node G. It assigns a unique private address to Node G and initiates a Route Request (RREQ) through broadcast. It further initializes an empty Route entry in its routing table for Node G. The transmitted packet of RREQ is shown with $I_{D1}=I_A$ $I_{D2}=I_G$ and $I_{D3}=I_A$ the I_{D4}=Sequence No. is used in our design this field is the number of RREQ messages sent by the node and it maintains this number. This field is used to avoid rebroadcast of RREQ by other nodes if once it has broadcasted the RREQ. In the data field we have the **PIP (A) =G (A) =10.129.5.2** the private address assigned by A in its private address map for Node G. The fields I_{D1}, I_{D2}, I_{D4} and Data remain constant in each transmission of RREQ packet where as I_{D3} is the intermediate transmitter ID and is changed to the ID of transmitter which broadcasts the packet. During each Broadcast, the Broadcasting Node initiates a non finalized nexthop entry back to the source node this entry is filled if it receives unicast Route reply RREP message. In order to ensure not broadcasting the message twice we can see that when the packet reaches the Node B it broadcasts the RREQ and records the broadcast ID which is the source node and its sequence number and it neglects the RREQ message when it receives it from C and D Nodes. Similarly, A also neglects the RREQ broadcasts from the Node B. When the packet reaches the destination Node G, the destination node G sends the unicast RREP to the first RREQ it receives it further neglects all the similar RREQ messages categorized by same Sequence Number of Source node A. The RREP process is shown in Fig.7. During the RREP process the each node sends back the RREP based on the values of its Routing table entry. It furthers make another entry for the destination node in its routing table based on the value of I_{D3} which it receives. e.g. Node F receives $ID_3= I_G$ so the next hop = I_G for $I_{G..}$ Similarly, Node C receives $ID_3=I_F$ so the nexthop for I_G is I_F and so on. When the RREP reaches the Source Node A the path between Source and Destination is establishes and Packets can flow between the source and Destination nodes.

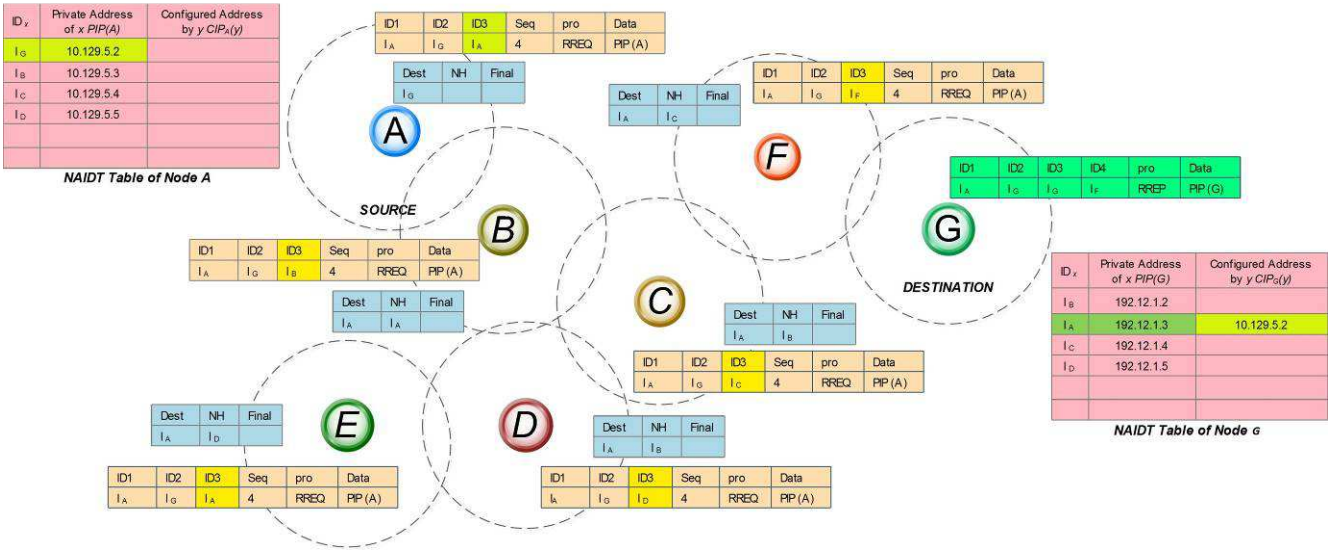

Figure 6. Route Request (RREQ) Process

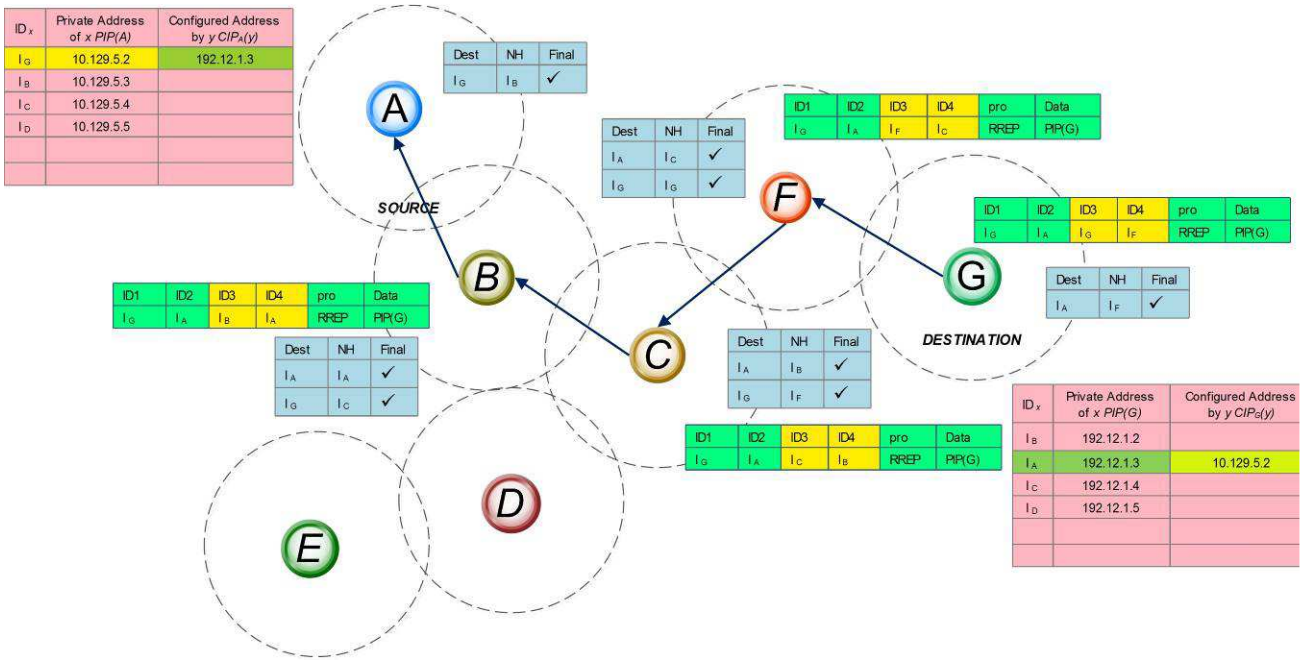

Figure 7. Route Reply (RREP) Process

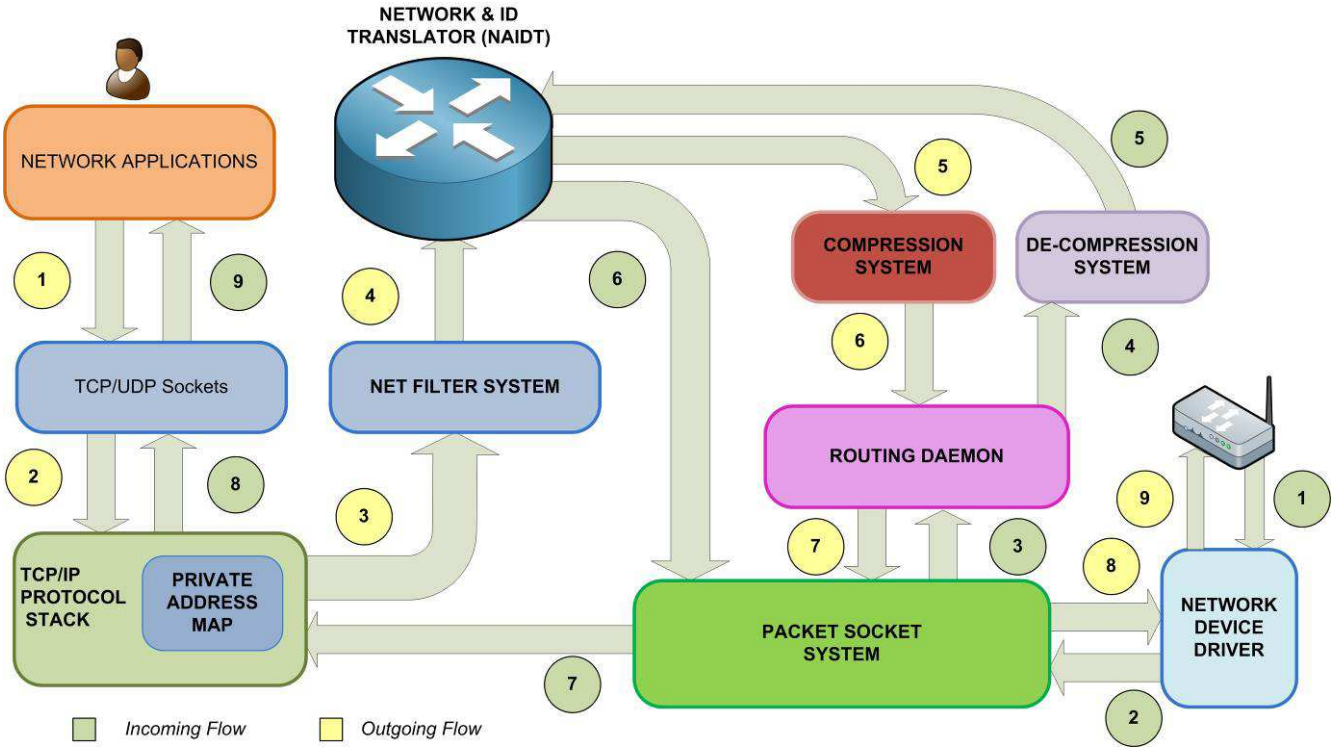

Figure 8. System Architecture of ID Based Auto-configuration Protocol with Process Flows

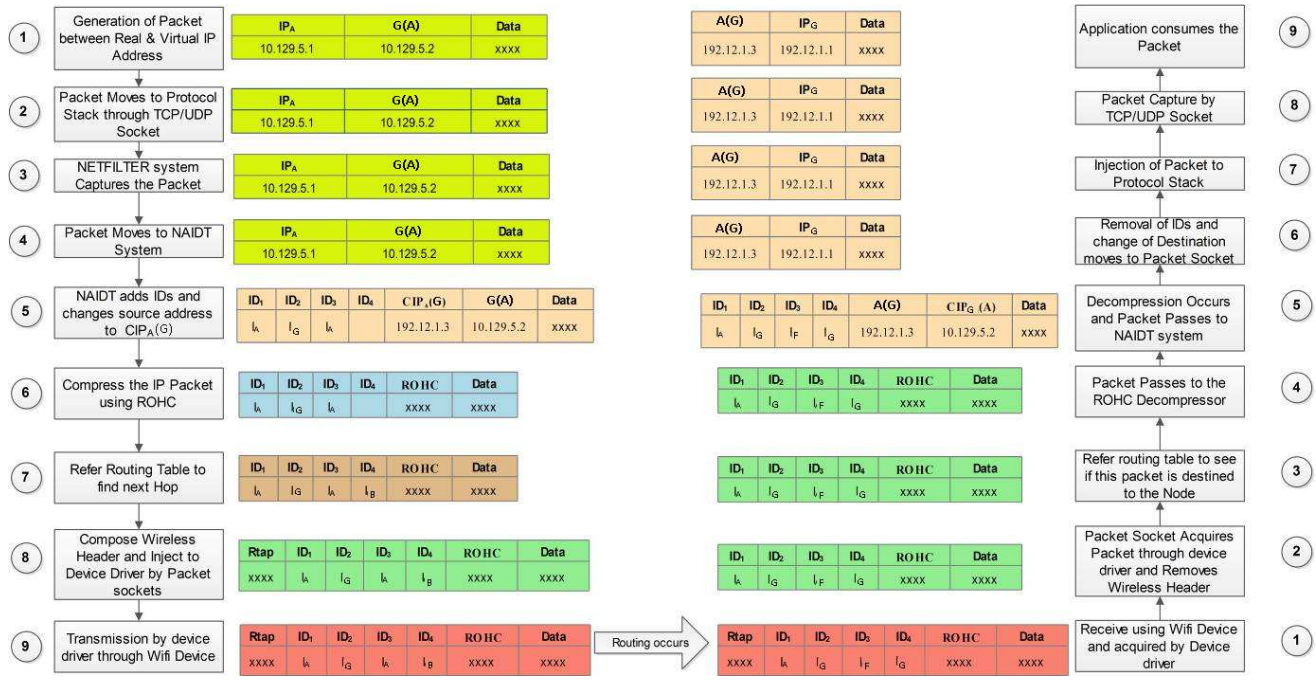

Figure 9. IP based Application Process Flows

5.2 End to End IP Based Application Process Flow

After the path discovery process the next step is to use the IP Based application with the system in integrated form is shown in Figure 8. whereas the steps of the flow is illustrated in Figure 9. The flow begins with **step 1** when node A generates the IP Packets and sends it to the TCP/UDP Socket the Socket captures the packet and feeds it to the TCP/IP Protocol stack in **step 2**. In **Step 3** the Net Filter system captures the packet from the protocol stack by using the NETFILTER hooks and sets verdict of dropping the packet from the protocol stack. In **Step 4** the packet moves to the NAIDT system. In **Step 5** the NAIDT system adds identifiers ID_1, ID_2 and ID_3 to packet which are source address, Destination address and the intermediate Source address ID_1 and ID_2 always remain constant throughout the journey of the Packet. After referring to the NAIDT table the NAIDT system also changes the source address to the configured IP address by G into A's table $CIP_A(G)$ the packet further moves to ROHC Compression system. The ROHC compressor compresses the packet. In step 7 routing table is referred for knowing the Next hop address to reach Node G and ID_4 is updated accordingly. In Step 8 Wireless Headers which include radiotap headers and necessary Wifi headers are appended to the packet and the packet socket injects the packet to the Wifi driver. The Wifi driver after receiving the packet transmits the packet in air through the Wifi interface. After the transmission the packet completes the journey hop by hop through intermediate nodes. There is no need to decompress the packet during each hop as the decision for next hop is based on identifiers and not on IP addresses. After reaching the Destination node G the packet is first received at the Wifi interface in **Step 1** the Packet socket system receives the packet from the Wifi driver in **Step 2** and removes the wireless headers of the packet. The routing table is referred in Step 3 to know whether the destination node has been reached or not. In **Step 4** ROHC Decompression is performed for recovering the IP Packet. In **Step 5** the packet is moved to the NAIDT system. The system changes the destination address as per the ID of node A and the packet becomes compatible to be injected to the Protocol stack. The packet moves to the Packet socket system in **step 6**. In **step 7** the packet socket system injects the packet to the protocol stack. The Packet is captured by the TCP/UDP socket in **Step 8**. The packet is consumed by IP based application in **Step 9**.

6. Testing and Implementation

We have verified various aspects of our proposed architecture through implementation and testing the protocol on real machines. The testing scenario is depicted in Figure 10. in which 03 machines are used in which machine A is the source and machine C is the destination node. Each node has a unique ID. When A wants to establish communication with Node C communication channel is established using underlying routing or path establishment mechanism. Instances of virtual Ethernet are run on both source and destination nodes and private address map is generated. Our protocol generates virtual Ethernet interfaces as per requirement. In this case when we send a request from Node A the protocol generates two in number virtual Ethernet devices at Node A. In the similar manner protocol also configures two in number virtual Ethernet interfaces at Node C. A depiction of virtual Ethernet interfaces configured at a Node is shown in Figure 11. When we issued ifconfig command at the terminal the virtual interfaces veth0 and veth1 are present which are shown in Figure 12. As a configuration step the system sets IP Table rules at both end points. For validating the plausibility of running IP based applications through our implemented protocol we have tested the protocol in two IP based applications which are regular network ping between Node A and Node B and IP based LAN chat application IPTux[47], the results and description is discussed in the ensuing paragraphs.

6.1 Ping Application

In order to verify the network connectivity between the end points ping is a popular application. We have tested the program to run between the nodes A and C. After the completion of path discovery phase when A wants to ping C it simply runs the ping application and issues command ping at terminal for IP Address 10.0.0.2. The IP traffic capture at the destination node is shown in Figure 13. which depicts proper reception of Ping request from node A and with operating system generated ID, Sequence number and the ping request number against each received ping request Node C generates a ping reply which is forwarded to Node A as a response.

6.2 Chat Application

In another test scenario when we run the IPTux application we can see the presence of two nodes in our network. We can initiate chat with the destination node in the same manner as we use normal LAN or IP based chat applications. When we type a message for the destination node and send it the message is successfully delivered at the other side. In a similar manner Node C can also send message to node A. The result for running IPTux between two MANET nodes is shown in Figure 14.

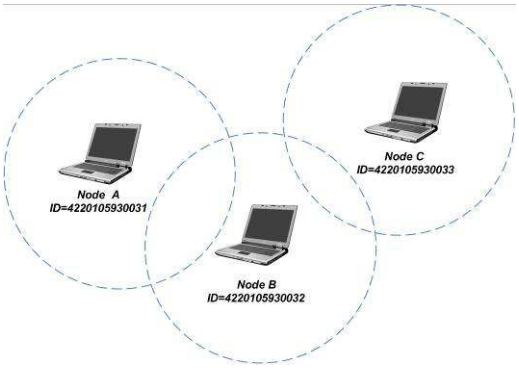

Figure 10. Experimental Setup

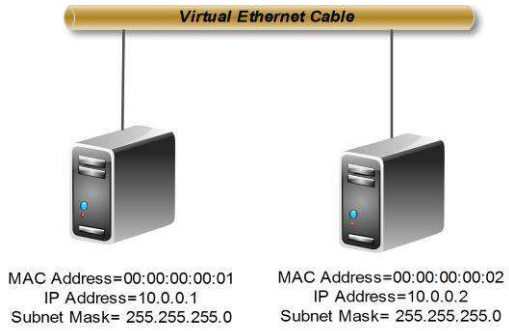

Figure 11. Visualization of Virtual Ethernet Network

```
veth0     Link encap:Ethernet  HWaddr 00:00:00:00:00:01
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::200:ff:fe00:1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:28 errors:0 dropped:0 overruns:0 frame:0
          TX packets:36 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5386 (5.3 KB)  TX bytes:6734 (6.7 KB)

veth1     Link encap:Ethernet  HWaddr 00:00:00:00:00:02
          inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::200:ff:fe00:2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:36 errors:0 dropped:0 overruns:0 frame:0
          TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6734 (6.7 KB)  TX bytes:5386 (5.3 KB)
```

Figure 12. Virtual Interfaces veth0 and veth1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0a1b, seq=1/256, ttl=64
2	0.000034	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0a1b, seq=1/256, ttl=64
3	0.998994	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0a1b, seq=2/512, ttl=64
4	0.999027	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0a1b, seq=2/512, ttl=64
5	1.998002	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0a1b, seq=3/768, ttl=64
6	1.998036	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0a1b, seq=3/768, ttl=64
7	2.997003	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0a1b, seq=4/1024, ttl=64
8	2.997035	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0a1b, seq=4/1024, ttl=64
9	3.996648	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0a1b, seq=5/1280, ttl=64
10	3.996685	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0a1b, seq=5/1280, ttl=64
11	4.996760	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0a1b, seq=6/1536, ttl=64
12	4.996791	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0a1b, seq=6/1536, ttl=64
13	5.012683	00:00:00_00:00:02	00:00:00_00:00:01	ARP	42	Who has 10.0.0.1? Tell 10.0.0.2
14	5.012710	00:00:00_00:00:01	00:00:00_00:00:02	ARP	42	10.0.0.1 is at 00:00:00:00:00:01

Figure 13. Ping Application between two MANET nodes

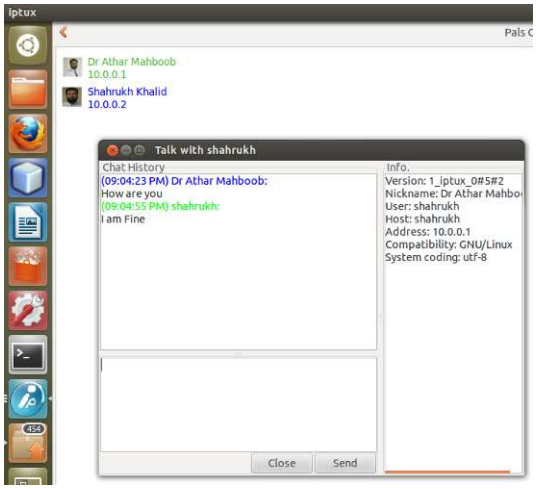

Figure 14. IPTux chat program running between MANET nodes

7. Qualitative Comparison of proposed Framework

Table II shows comparison of our proposed ID based protocol with current auto-configuration protocols on the basis of various attributes. Due to the usage of real world identifiers with a virtual Ethernet configuration the protocol can ensure no IP address conflicts. Support of ROHC can be beneficial with respect to efficiency especially for streaming applications like VOIP. Other benefits include no management of partitioning merging and no requirement for address reclamation.

Table II. Qualitative Comparison of ID Based and current auto-configuration protocols

ATTRIBUTE	PROPOSED ID BASED AUTOCONF PROTOCOL	IP BASED AUTOCONF PROTOCOLS
IP Address conflicts	Does not suffer from IP address conflicts due to privately managed IP address maps.	**Suffer from IP address conflicts of varying degrees especially during the course of nodes merging.**
Efficiency provision	Uses ROHC mechanism for IP header compression	**Do not support ROHC as routing is done based on IP addresses.**
Communication Overhead	Uses ROHC mechanism and do not require extra protocol messages to maintain unique IP addresses throughout MANET.	**Requires exchanging various messages for maintaining unique IP addresses.**
Scalability	Highly scalable as do not require generating messages through out MANET for assuring unique IP addresses.	**Various protocols suffer in scalability of varying degrees due to messages exchanging.**
Uniqueness	Ensures uniqueness based on unique Identifiers like national identity card, telephone number etc.	**Uses underlying algorithm for maintaining unique IP address throughout MANET.**
Allocation Latency	Minimal and only requires initial path discovery.	**Varying degree of allocation latency for configuration of unique IP addresses.**
Partitioning and Merging	Do not require to handle partitioning and merging problems	**Various strategies for handling partitioning and merging are required.**
Reliability	Reliable communication can be ensured as no breaks in communication will happen due to IP address variations.	**Different protocols suffer in reliability of varying degrees.**
Address reclamation	No need for ensuring address reclamation strategy as a node maintains only private address map.	**Address reclamation strategy is required.**
Heterogeneity	All nodes can maintain heterogeneous IP address schemes.	**Heterogeneity is a problem and nodes need to come to a uniform IP address scheme during the process of merging.**
Complexity	Less complex protocol design using clear message flows and least development effort.	**Difficult to ascertain complexity as development efforts are not carried out.**
Security provision	HIP[48] or EMILSA [24] like security provisions can be worked out. However rigorous research is required to be carried out for security related implementation.	**IP based security provisions can be implemented. Least amount of research has been carried out in this direction.**
Uniformity	**All nodes have same roles.**	**Various protocols have different strategies some are uniform and some are not.**

8. Conclusion and Future Work

In this paper we have demonstrated the design and implementation of an ID based MANET auto-configuration protocol for solving the IP based auto-configuration issues in MANET context. The implemented design uses identifiers like national identity card number, vehicle number, student ID card number for establishing communication between end points. An ID based routing and packet forwarding mechanism is also designed and implemented. The implemented architecture supports conventional IP based applications and users can run these applications in MANET scenario as if the MANET nodes are present in a LAN based environment. We have demonstrated two applications which are Ping and an IP based LAN chat program IPTux between two MANET nodes. The protocol supports the use of ROHC compression which is another benefit in terms of efficient service provisioning. In Future we will concentrate on data analysis of data obtained from ROHC based compression. We will also perform rigorous testing of the ID based routing daemon. Further we will work on security related issues for our proposed architecture.

9. Acknowledgement

We want to thank GEC members of Hamdard University for their guidance and support during the conduct of this doctoral research of Corresponding Author Shahrukh Khalid. Furthermore we are thankful for the anonymous reviewer for providing valuable comments and suggestions for improving the quality of this paper.

References

[1] R. Droms, “Dynamic host configuration protocol,” RFC 2131, Mar. 1997.

[2] O. Troan, R. Droms, “IPv6 Prefix Options for DHCPv6”, RFC 3633, December 2003. [Online], Available:http://www.ietf.org/rfc/rfc3633.txt

[3] S. Thomson,T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December, 1998

[4] T. Narten, E. Nordmark, W. Simpson, H. Soliman “Neighbor Discovery for IP version 6 (IPv6)”, RFC 4861 [Online], Available: http://www.ietf.org/rfc/rfc4861.txt

[5] L. Villalba, J.G. Martinez, A. Orozeo and J. Diaz, “Auto-configuration Protocols in Mobile ad hoc networks”, Sensors, pp 3652-3666, 2011.

[6] N. Wangi, R. Prasad, M. Jacobsson ,I. Niemegeers, “Address Autoconfiguration in wireless ad hoc networks:protocols and techniques”, IEEE Wireless Comm, pp 70-80,2008.

[7] H. Zhou ,M. Mutka, “Review of Autoconfiguration for MANETs”, Intech Review, 2012.

[8] R. Jain, “Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation”, Proceedings of Military Communications Conference (MILCOM), Washington, DC, pp 23-25, 2006.

[9] IEEE 802.11, “Wireless Local Area Networks” [Online], Available: http://www.ieee802.org/11/

[10] B. Brownlee Y. Liang, “Mobile Ad HocNetworks An Evaluation of Smart phone Technologies”, Defence R&D Canada , DRDC CORA CR 2011-169,2011.

[11] J. Cano, J-C. Cano, C. Calafate, P. Manzoni, “EasyMANET: an extensible and configurable platform for service provisioning in MANET environments”Communication Magazine, IEEE, 2010.

[12] J. Cano, "Integrated Architecture for Configuration and Service Management in MANET Environments", PhD Thesis,Valencia, 2012.

[13] L. Villalba, J. Matesanz, A. Orozco,A. M. Díaz, “Distributed Dynamic Host Configuration Protocol (D2HCP)”,Sensors, 2011.

[14] L. Villalba, J. Matesanz, A. Orozco ,J. Cortez, “An extension proposal of D2HCP for network merging”, Journal of Ubiquitous & Pervasive Networks, Vol. 3 No. 1,pp 35-40,2011.

[15] M. Al-Mistarihi, M. Al-Shurman, A. Qudimat, “Tree based dynamic address auto-configuration in mobile ad hoc networks”, Elsevier, Computer Networks, Vol 55, pp 1894-1908, 2011.

[16] U. Ghosh, R. Datta, “A secure dynamic IP configuration scheme for mobile ad hoc networks”, Ad Hoc Networks, Elsevier, vol 9, pp 1327-1342,2011.

[17] U. Ghosh, R. Datta, “An ID based secure distributed dynamic IP configuration scheme for mobile ad hoc networks”, ICDCN'12 Proceedings of the 13th international conference on Distributed Computing and Networking,Springer, pp 295-308.,2012.

[18] Z. Slimane, A. Abdelmalek,M. Feham ,A. Taleb-Ahmed,, “Secure and robust IPV6 autoconfiguration protocol for mobile adhoc networks under strong adversarial model”, International, Journal of Computer Networks & Communications (IJCNC) Vol.3, No.4, 2011.

[19] A. Abdelmalek, Z. Slimane, M. Feham, A. Taleb-Ahmed TCSAP, , “A New Secure and Robust Modified MANETconf Protocol”, WiMo/CoNeCo Springer, CCIS 162, pp. 73–82, 2011.

[20] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, S. Surana, Internet Indirection Infrastructure, IEEE/ACM Transaction on Networking 12 (2) pp 205–218,2004.

[21] Merriam Webster [Online], Available: http://www.merriam-webster.com

[22] J. Pan,R. Jain, S. Paul, "MILSA: A New Evolutionary Architecture for Scalability, Mobility, and Multihoming in the Future Internet", IEEE journal on selected areas in communications, Vol. 28, NO. 8, pp 1344-1362, 2010.

[23] J. Pan, S. Paul, R. Jain, M. Bowman, “MILSA: A Mobility and Multihoming Supporting Identifier-Locator Split Architecture for Naming in the Next Generation Internet”, Globecom ,2008.

[24] J.Pan, S. Paul, R.Jain, M. Bowman, S. Chen, “Enhanced MILSA Architecture for Naming, Addressing, Routing and Security Issues in the Next Generation Internet”, Proceedings of IEEE (GLOBECOM), pp. 1-6, 2009.

[25] C. So-In, R. Jain, S. Paul ,J. Pan, "Virtualization architecture using the ID/Locator split concept for Future Wireless Networks (FWNs)",Journal of Computer Networks 55, Elsevier, pp 415–430,2011.

[26] V. Kafle , M. Inoue, “HIMALIS: Heterogeneity inclusion and mobility adaptation through locator ID separation in new generation networks”, IEICE Trans. on Commun., Vol. E93-B, No. 3, pp. 478-489, 2010.

[27] Y. Wang, J. B. Xiaoke,”Mobility Support in the internet Using Identifiers”, CFI’12, ACM, September, 2012.

[28] P. Martinez-Julia, A. Gomez, “A Novel Identity based Network for Next Generation Internet”, Journal of Universal Computer Science,Vol 18,pp 1643-1661,2012.

[29] Y.Wang, J. Bi, C.Peng, H. Hu, “UNA:A new Internet Architecture for User level multihoming and mobility”,CF’11,ACM,2011.

[30] A. Boukerche, “Performance Evaluation of Routing Protocols for Ad Hoc Wireless Net-works,” Mobile Networks and Applications, vol. 9, no. 4, pp. 333–342, 2004.

[31] E. Borgia, “Experimental Evaluation of Ad Hoc Routing Protocols”, Proceedings of IEEE International Conference on Pervasive Computing and Communications Workshops, 2005.

[32] H. Wang, “A Robust Header Compression Method for Ad hoc Network”, IEEE, 2006.

[33] S. Gowrishankar, T. Basavaraju, S. Kumar Sarkar, “Analysis of Overhead Control Mechanisms in Mobile AD HOC Networks”,Lecture Notes in Electrical Engineering, Springer Science+Business Media B.V. 2009.

[34] Library Robust Header Compression, [Online]. Available: https://launchpad.net/rohc

[35] W. Ang, T. Wan, K. Kotaoka ,C. Teh, "Performance Evaluation of Robust Header Compression (ROHC) over unidirectional Links using DVBS Test beds,KEIO SFC JOURNAL Vol 8 No.2,2008.

[36] Y.Wey Chong, T. Wan, Header compression scheme in point-to-point link model over hybrid satellite Wimax network, International Journal of Ad hoc Sensor and Ubiquitous Computing (IJASUC), Vol 2 No.03, 2011.

[37] Ubuntu 12.04 LTS [Online], Available:https://wiki.ubuntu.com/PrecisePangolin/ReleaseNotes/UbuntuDesktop-12.04.1.

[38] A.Tudzarov ,T.Janevski, “Design for 5G Mobile Network Architecture", International Journal of Communication Networks and Information Security (IJCNIS), Vol. 3, No. 2, 2011.

[39] M. Vipin ,S. Srikanth, “Analysis of open source driver for IEEE 802.11 WLANs”, Wireless Communication and Sensor Computing ICWCSC, 2010.
[40] Libnetfilter, [Online]. Available: http://www.netfilter.org/projects/libnetfilter_queue
[41] Libnfnetlink, [Online]. Available: http://www.netfilter.org/projects/libnfnetlink
[42] IPTABLES, [Online]. Available: http://www.netfilter.org/projects/iptables/
[43] M. Kerrisk,"The Linux Programming Interface",No Starch Press,ISBN:1593272200,2010
[44] C. Perkins, E. Belding-Royer , S. Das, “Ad hoc on demand distance vector (AODV) routing”, IETF RFC 3561, July 2003.
[45] I. Chakeres, L. Klein-Berndt, “AODVjr, AODV simplified”, ACM SIGMOBILE Mobile Computing and Communications Review, pp. 100–101, 2002.
[46] H. Zafar, N. Alhamahmy, D. Harle ,I. Andonovic, "Survey of Reactive and Hybrid Routing Protocols for Mobile Ad Hoc Networks",International Journal of Communication Networks and Information Security(IJCNIS),Vol. 3, No. 3, 2011.
[47] IPTux, [Online]. Available:https:github.com/iptux-src/iptux
[48] R. Moskowitz , P. Nikander, “Host Identity Protocol architecture”, IETF RFC 4423, May 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4423.txt