IDHOCNET-A Novel Protocol Stack and Architecture for Ad hoc Networks

S. Khalid[1], A. Mahboob[2], F.Azim[3] and A.Rehman[4]

[1,3,4,] Hamdard University Karachi, Pakistan
[2] DHA Suffa University Karachi, Pakistan
shahrukh_khalid@hotmail.com,dean@dsu.edu.pk, fahad.azim@hamdard.edu,aqeel.rehman@hamdard.edu

Abstract: Presently employed Internet Protocol (IP) stack possesses number of architectural problems. The contemporary research direction for the improvement of present Internet architecture mainly focuses on the use of real identifiers instead of IP addresses for host identification in the network. However, the proposed architectures mostly discuss the infrastructure oriented network models and minimal research has been conducted in the direction of proposals for ad hoc networks. In order to resolve the present limitations of ad hoc networks, we describe an implementation of a novel identifier based ad hoc network protocol stack and architecture known as IDHOCNET (Identifier based ad hoc network). The architecture proposes a novel paradigm of identifier based applications for multi-hop wireless ad hoc environment. The proposed system further provides backward compatibility to support co-existence with IP based applications. As a proof of concept, the architecture has been implemented on Linux platform with WiFi interfaces. Various practical scenarios with architectural insight are presented to demonstrate the practicability of the proposed approach.

Keywords: Identifier based networking, IP-less networks, Wireless ad hoc networks, Identifier based applications, Private IP addresses, ID/Locator Split Architectures, WiFi interface.

1. Introduction

Current Internet Protocol (IP) based networking model was proposed almost four decades ago [1]. Since its inception, there has been unprecedented growth of Internet. The core architecture of Internet is based on assumption of wired networks with infrastructure support. The design of Internet is deeply embedded in Internet Protocol (IP) address. IP address has a topological bearing and it is assigned by Internet Assigned Numbers Authority (IANA)[2].

In order to ensure proper functioning of the network of such an enormous size a number of infrastructure component are integrated in the architecture. These include Domain Name Systems (DNS) [3], Dynamic Host Control Protocol (DHCP) [4], Network Address Translator (NAT) [5], etc. In the Internet, DNS servers are placed across the globe to resolve the IP addresses for the named entities which the user seeks. However, after the resolution of name to corresponding IP, the regular Internet traffic always flows on the basis of IP addresses. The current Internet implementation possesses number of issues and researchers have provided critique on the architectural problems of Internet [6]-[7]. These problems include Multihoming, Dual/Overriding role of IP address as identifier of the host and locator of the host. The structure of the IP address as an identifier is also a debatable aspect. IP address is assigned when a host joins the network. When the host moves to another network all the previous connections or associations of previous point of attachment are lost. When the Internet based architecture is applied to the ad hoc networking scenario the problems increase manifold. In ad hoc networking environment nodes are placed wirelessly and their positions are not fixed. Moreover, there is no support of infrastructure components like DNS, DHCP or NAT servers in ad hoc network settings. The biggest problem of ad hoc networks is the IP address auto configuration problem. There are numerous proposals which are based on different mechanisms for ensuring unique assignment of IP addresses in all nodes of the network [8]-[9]. Moreover, naming of the nodes in a distributed manner is also problematic [10]. A figurative representation to the problems of ad hoc networking realized by IP centric approach is depicted in Figure 1.

The core problems inherent to the present architecture instigated the Internet architecture research. The main direction of the proposals is derived from the pioneer work of Stoica et.al [11]. They suggested an Internet Indirection Infrastructure (I^3) strategy in which each entity is identified by its identifier. I^3 uses a special gateway for indirection of the packets to appropriate locations in the networks. The research area is now categorized as ID/Locator Split concept. Rigorous reviews of ID/Locator based architectures is available in [12] and [13]. Among large number of ID/Locator split architectures popular architectures include Heterogeneity Inclusion and Mobility Adaptation (HIMALIS) [14], Mobile Oriented Future Internet (MOFI) [15], Mobility and Multihoming supporting Identifier Locator Split Architecture (MILSA) [16], Enhanced Mobility and Multihoming supporting Identifier Locator Split Architecture (EMILSA) [17], Domain Insulated Autonomous Network Architecture (DIANA) [18] etc.

It is important to appreciate that the architectures are proposed for infrastructure based networks and cannot be directly used for ad hoc networking scenarios. Very little research has been conducted for adaptation of ID/Locator Split concept to the ad hoc networking case. In [19] authors provide a ID/Locator mapping system for the mobile ad hoc network whereas in [20] a routing strategy based on ID/Locator split is proposed. The works cited above are mainly focused on analytical models and simulation studies and very limited work has been carried out in the direction of a practical prototype ready to use by others.

In order to address the above problems related to the inherent problems of IP based networks with reference to its application in ad hoc networking, we propose an alternative protocol stack and architecture. Instead of adapting the simulation based strategy for verification our approach is based on the working prototype of the system. For the plausibility of the system in multi hop wireless ad hoc network settings the developed software was deployed on number of nodes and various network parameters were collected.

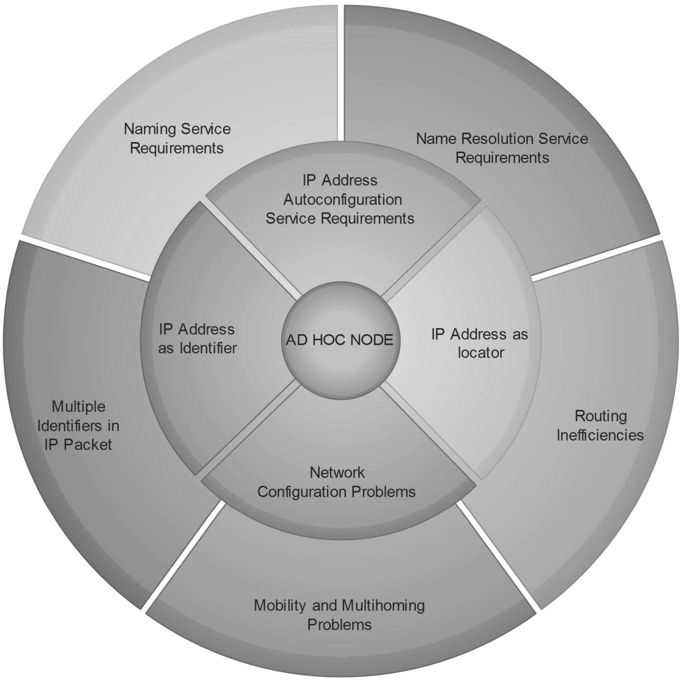

Naming Service Requirements
Name Resolution Service Requirements
IP Address Autoconfiguration Service Requirements
IP Address as Identifier
AD HOC NODE
IP Address as locator
Multiple Identifiers in IP Packet
Routing Inefficiencies
Network Configuration Problems
Mobility and Multihoming Problems

Figure 1. Problems of IP Centric realization of ad hoc networks

Moreover, alternative to IP based application, a novel Identifier based light weight applications design paradigm with practical applications is also presented in this paper. The prototype is built on Linux platform with open source application programming interfaces. The detailed implementation is given in the ensuing paragraphs.

The remainder of the paper is organized as follows: Section 2 gives practical design issues and problems of IP centric applicability for ad hoc networking case. Section 3 provides the system design details of the identifier based ad hoc protocol stack and architecture. Section 4 provides implementation insight of proposed architecture based on Linux platform. Section 5 provides the process flows of the implemented prototype. Section 6 provides results and discussion of various test performed on the prototype of the identifier based system. Finally Section 7 concludes the paper.

2. Design considerations and related issues

There are number of issues inherent in the IP centric approach when it is applied to the ad hoc networking domain. In the following paragraphs an insight of such problems is given.

2.1 IP Address Auto-configuration problems

One of the major problems of IP centric ad hoc network approach is IP address uniqueness problem. There are numerous IP address auto configuration protocols. Some latest IP address auto configuration proposal include filter based auto configuration [21], Ensemble based approach [22] and Auto configuration for 6LoWPAN [23] and cluster based approach [24]. IP Address auto configuration protocol or service must run to ensure unique IP addresses of each peer. As an example a real implementation of MANET known as High Level MANET Protocol (HLMP) API, uses strong and weak duplicate address detection (DAD) mechanism [25]-[26] for Auto configuration support.

The use of Auto configuration adds complexity to the overall software stack and further adds communication overhead. The use of IP Address Auto configuration can be avoided if IP addresses are not used as an identifier. Identifiers like MAC addresses can be converted in the form of identifiers as shown in Figure 2. IMSI, IMEI, telephone or mobile numbers can also be used as an identifier. Our work of identifier based auto configuration [27], uses identifiers for packet forwarding and path establishment purposes. As all types of packets including path establishment, IP based application data, ID based application data are forwarded on the basis of real identifiers, IP address auto configuration service is not required. In the proposed system IP addresses based applications are implemented by using private address map available at each node. At the receiving side the data is injected to the TCP/IP protocol stack of the node.

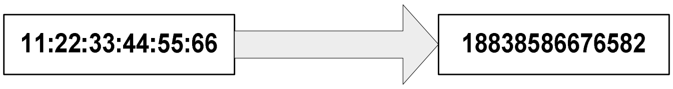


Figure 2. MAC Address to MAC ID conversion

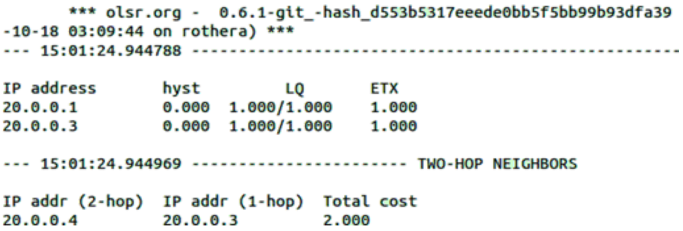
2.2 Locators in ad hoc networks

In Internet IP address holds hierarchical information. Around the globe Internet Assigned Numbers Authority (IANA) assigns the IP addresses to different networks located in various parts of the world. IANA assigns these addresses to Regional Internet Registries (RIRs) situated around the globe, where each RIR further assigns the allocated addresses to the Internet Service providers in their respective regions. In the present scenario of IP based ad hoc network settings. Each host is assigned an IP address, however due to the weak topology in ad hoc networks and highly dynamic topology in case of Mobile ad hoc networks the topology orientation of IP address losses its meanings.

As an example shown in Figure 3a, OLSR[28] is running on a number of ad hoc hosts at time T_o. At the stated snapshot a node identified by 20.0.0.1 has 20.0.0.3 as its one hop peer and 20.0.04 as its two hop peer. The node can reach its two hop peer by the help of its one hop peer. At later time say T_1 as shown in Figure 3b, due to the dynamicity of the network one hop and two hop peers of the hosts are changed. It can be appreciated that IP addresses in such cases have no role on the basis of their structure settings to get to another peer. Thus the role of IP address as a locator in infrastructure network cannot be fulfilled in ad hoc networking scenarios.

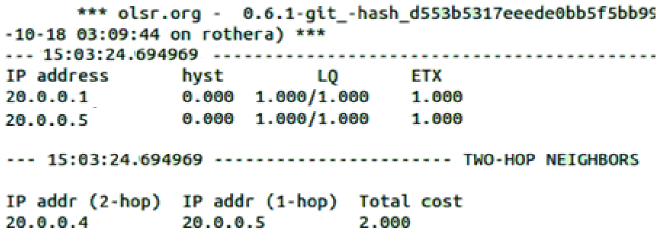
It is pertinent to mention 802.11s [29] wireless mesh networking standard(Figure 3c) in which routing is based on MAC addresses. But due to the requirement of IP based identification, a particular host is only identified on the basis of a unique IP address. Further all applications are based on IP based stack. Therefore each host possesses an IP to MAC mapping in their network stack.

Considering these cases, a flat identifier is more suitable which could identify the node and further route the packet to the destination on the basis of the next hop identifier.



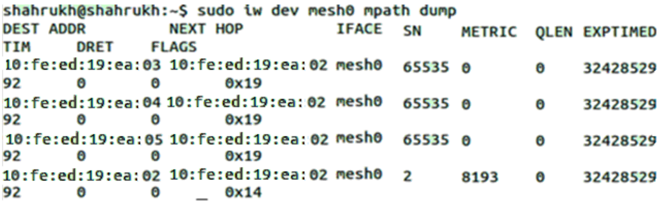
```
      *** olsr.org -  0.6.1-git_-hash_d553b5317eeede0bb5f5bb99b93dfa39
-10-18 03:09:44 on rothera) ***
--- 15:01:24.944788 ----------------------------------------------------

IP address       hyst         LQ        ETX
20.0.0.1         0.000  1.000/1.000     1.000
20.0.0.3         0.000  1.000/1.000     1.000

--- 15:01:24.944969 ---------------------- TWO-HOP NEIGHBORS

IP addr (2-hop)  IP addr (1-hop)  Total cost
20.0.0.4         20.0.0.3         2.000
```

Figure 3a. OLSR running at time T_o



```
      *** olsr.org -  0.6.1-git_-hash_d553b5317eeede0bb5f5bb9S
-10-18 03:09:44 on rothera) ***
--- 15:03:24.694969 -------------------------------------------
IP address       hyst         LQ        ETX
20.0.0.1         0.000  1.000/1.000     1.000
20.0.0.5         0.000  1.000/1.000     1.000

--- 15:03:24.694969 ---------------------- TWO-HOP NEIGHBORS

IP addr (2-hop)  IP addr (1-hop)  Total cost
20.0.0.4         20.0.0.5         2.000
```

Figure 3b. OLSR running at time T_1



```
shahrukh@shahrukh:~$ sudo iw dev mesh0 mpath dump
DEST ADDR          NEXT HOP           IFACE  SN     METRIC  QLEN EXPTIMED
TIM      DRET     FLAGS
10:fe:ed:19:ea:03 10:fe:ed:19:ea:02 mesh0  65535 0        0    32428529
92       0        0       0x19
10:fe:ed:19:ea:04 10:fe:ed:19:ea:02 mesh0  65535 0        0    32428529
92       0        0       0x19
10:fe:ed:19:ea:05 10:fe:ed:19:ea:02 mesh0  65535 0        0    32428529
92       0        0       0x19
10:fe:ed:19:ea:02 10:fe:ed:19:ea:02 mesh0  2     8193     0    32428529
92       0        0     _ 0x14
```

Figure 3c. 802.11s established path view

Figure 4 shows the routing tables of an identifier based path establishment scenario. In the available scenario every node can send packets on the basis of identifier of the peer.

2.3 Identification in ad hoc settings

In Internet, IP address holds the role of identifier of the communication entity. The structure of an IPv4 address is based on the dotted decimal notation having 04 decimal blocks of 01 byte each. More extended version of IP address IPv6 is 4 times larger and represented in the form of hexadecimal numbers. Figure 5 illustrate structure of IPv4 and IPv6 IP addresses.

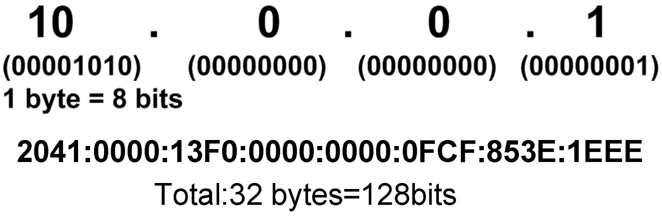


Figure 5. Structure of IPv4 and IPv6 addresses

The structure of IP address does not qualify as a friendly identifier of a host. The identifier is difficult to remember. In infrastructure based networks like Internet entities are given fully qualified domain names (FQDN). Domain Name System (DNS) servers are used for providing the Name to IP mapping services.

The mapping service is manageable to achieve in the infrastructure oriented networks. However, in order to ensure the ease of use naming and name resolution services in ad hoc networks distributed naming and name resolution service is difficult to achieve. Naming and Name resolution services further add complexity in the overall software stack of the host. Moreover there is considerable addition of communication overhead after such services are in operation.

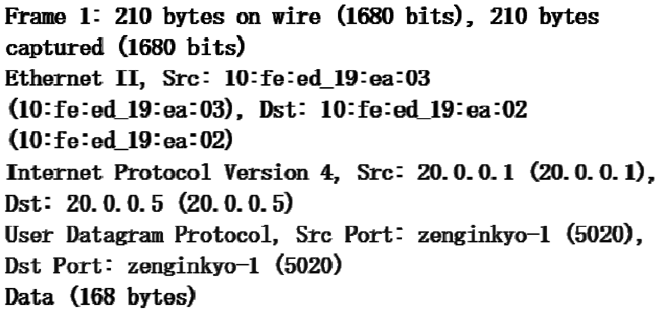
An alternative approach for node identification is through the use of proper identifiers like MAC ID, Global telephone or mobile numbers, IMSI, IMEI etc. When such identifiers are used the naming and name resolution services are not required. Therefore the design of the overall system will become simplified. Moreover considerable amount of networking bandwidth can be saved.

2.4 Multiple identifiers in a packet

In an IP based ad hoc networks a transmitted packet contains different identifiers. Figure 6 shows the captured frame by wireshark in ad hoc network settings when an OLSR based network and 802.11s network were configured. It is pertinent to note that same type of frame is received at the destination node.

The received frame contains source and destination MAC address and as well as source and destination IP addresses.

In the proposed system i.e. IDHOCNET, only a single type of Source and Destination identifiers are used.



```
Frame 1: 210 bytes on wire (1680 bits), 210 bytes
captured (1680 bits)
Ethernet II, Src: 10:fe:ed_19:ea:03
(10:fe:ed_19:ea:03), Dst: 10:fe:ed_19:ea:02
(10:fe:ed_19:ea:02)
Internet Protocol Version 4, Src: 20.0.0.1 (20.0.0.1),
Dst: 20.0.0.5 (20.0.0.5)
User Datagram Protocol, Src Port: zenginkyo-1 (5020),
Dst Port: zenginkyo-1 (5020)
Data (168 bytes)
```

Figure 6. Multiple Identifiers in a packet

3. Architectural Details

Architecture of the proposed IDHOCNET system is presented in the ensuing paragraphs. A comparison of IDHOCNET with the available IP based ad hoc network approach is shown in Table 1.

3.1 Identifiers of the proposed system

In real world a person assumes different roles and moves to various locations to perform different activities as shown in Figure 7. In every such scenario a person assumes different identification and holds different identifiers, like living in a home a person is identified by a name. In the scenario of university there is unique student identity card number, like 1234-98-99 which signifies roll number 1234 of 1998/99 batch. In an office an employee number is allocated, like ABC-123 where ABC is the company name and 123 is the serial number. Living in a country a national identity card number is assigned to a person, like 42201-XXXXX111-4. If a global presence is required there are established means like global telephone and cell phone numbers. Moreover, a number of globally unique identifiers of the devices like MAC Addresses, IMSI and IMEI numbers are common. There exist different realms in which a person can work and an ad hoc network can be formed in each of such realm. Necessary security can be associated with a particular realm. In the proposed id based architecture a person will be required to use a unique identifier for communication and identification in a particular realm.

3.2 Identifiers Based Communication

An identifier is a unique entity in the context of a realm in which a particular ad hoc network has been established. In the strict sense, the identifier in the proposed architecture is hardware ID installed apriori in a host. Such a hardware id can be translated in packet headers for establishing the means of communication. As an example, the architecture uses MAC address as an Identifier as it is easily translatable to an ID. After translation it can be observed that the identifier looks similar to a telephone number which is remembered or can be written down in the directory for communicating with the desired party. There are other hardware based identifiers like SIM number, IMSI, IMEI etc. Moreover there is a current prevalent of RFID technology which also holds an identifier. The technology can be used for embedding the identifiers like national identity card number, telephone number, employee number, home member number etc after permission from a respective authority which owns the identifier. As shown in Figure 8, such identifiers can be realized in the form of contact number style index which is a common practice in today’s world.

IDENTIFIER	NAME
188385866765X2	Asad
188385866765X3	John
188385866765X4	Amir
188385866765X5	Khan

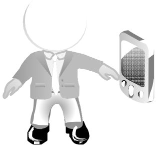
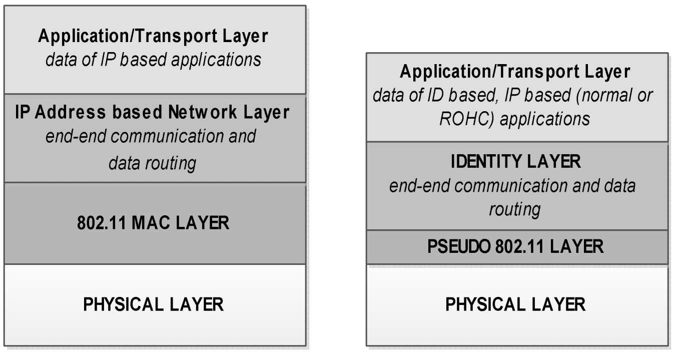


Figure 8. Identifiers contacts

3.3 Protocol stack of identifier based network

As compared to regular IP based architecture which uses a full 802.11 MAC header, IDHOCNET uses a minimal pseudo 802.11 header of 04 bytes. Figure 9 a. shows the IP based protocol stack used by 802.11 based MAC and the proposed IDHOCNET architecture in Figure 9 b. The end to end communication between the entities in IDHOCNET is achieved by using the identity layer instead of IP layer. The protocol stack supports a novel type of applications which can be realized in IDHOCNET. These are identifier based network applications.

In the proposed system all the data whether routing or application based is transmitted on the basis of identifiers. The support for contemporary IP based applications is provided by the private IP address map which is maintained by each node of the IDHOCNET system. The detail of IP based application provisioning is given in the ensuing paragraphs. Another aspect which is achieved in the system is by running all IP based applications by the identifier of the remote host. This aspect is further covered in section 5.2.1


Application/Transport Layer
data of IP based applications
IP Address based Network Layer
end-end communication and data routing
802.11 MAC LAYER
PHYSICAL LAYER
Application/Transport Layer
data of ID based, IP based (normal or ROHC) applications
IDENTITY LAYER
end-end communication and data routing
PSEUDO 802.11 LAYER
PHYSICAL LAYER

a. IP based protocol stack b. IDHOCNET protocol stack

Figure 9. Protocol stacks for IP based architecture and proposed IDHOCNET

3.4 Private IP Address Map

The protocol stack supports IP based applications through the use of private IP address map implemented locally at each node. As shown in Figure 10, a view of private IP address map at an IDHOCNET node is shown. A virtual interface is initialized in the protocol stack. Whenever the node requires IP based communication with another node a private IP address is added to the map and a synonym of the interface is initialized. Furthermore a filter rule is added by IDHOCNET for all the outgoing traffic when user runs an IP based application on the respective IP address configured for the remote peer. Upon capture of the outgoing IP packet in IDHOCNET system, the respective ID is obtained from the network table and an ID based packet is transmitted from the system.

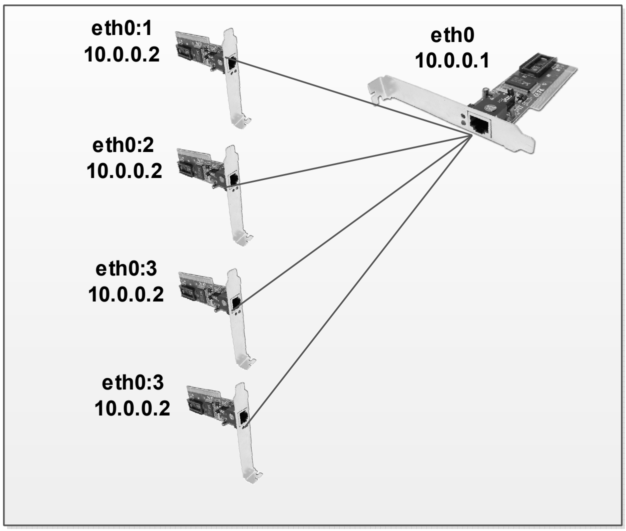


Figure 10. Private IP Addresses internal to the IDHOCNET host

3.5 Frame Formats

IDHOCNET uses ID based frames for application data exchange and communication establishment between the peers. Figure 11. shows the basic frame types of IDHOCNET. For every message to transmit through the WiFi interface a pseudo 802.11 header of 04 bytes is added before the ID based header.

A realm identifier is added which indicates the domain in which a particular ad hoc network has been formed. A realm can be formed in which all users are communicating with each other by using a particular type of identifier. As an example, a realm can be formed in which all users use their MAC ID, in other setting users can use their global telephone numbers, etc.

Following are the message types of IDHOCNET:

- RREQ packet
- RREP packet
- Data packet

Other type of frames can be added to the system for performing other type of operations.

The general processing of the frame when received at the Wifi interface is illustrated at the algorithm at Figure 12.

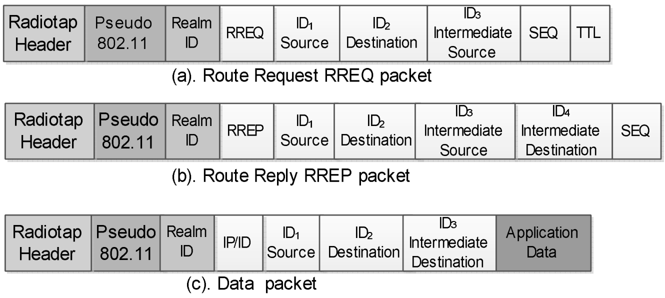
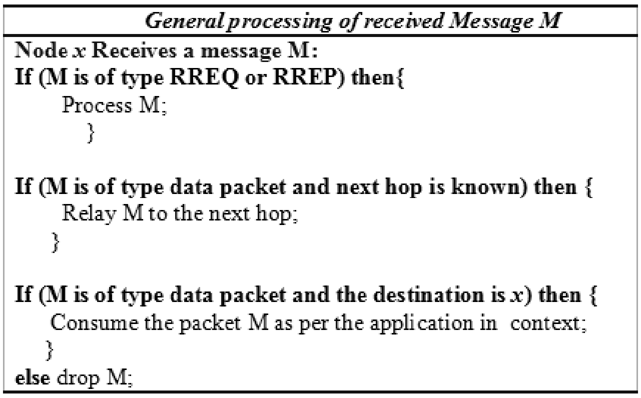


Figure 11. Packet types of IDHOCNET



```
General processing of received Message M
Node x Receives a message M:
If (M is of type RREQ or RREP) then{
    Process M;
      }

If (M is of type data packet and next hop is known) then {
    Relay M to the next hop;
  }

If (M is of type data packet and the destination is x) then {
   Consume the packet M as per the application in context;
  }
else drop M;
```

Figure 12. General packet processing of IDHOCNET

3.6 ID based application design

An Inter Process Communication (IPC) is a mechanism through which applications can exchange data. As shown in Figure 13, an ID based application is designed by using an Inter Process Communication (IPC) facility of the system. Such a mechanism must not depend on IP based protocol stack.

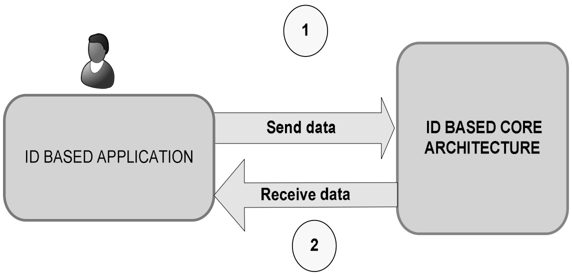


Figure 13. Data exchange between IDHOCNET and ID based application

4. Implementation Details

4.1 Sockets APIs of IDHOCNET

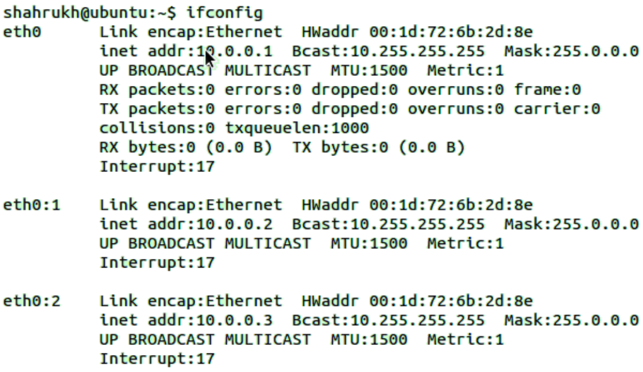
IDHOCNET uses various sockets API for achieving its functionalities. The interfaces of IDHOCNET with sockets API is illustrated in Figure 14. The function of sockets API used by IDHOCNET is listed below:

- Netlink Sockets [30] for capturing the IP based packets
- Packet Sockets [31] for injection and capturing of packets from Wifi interface
- Local Name Space Socket [32] for implementing ID based application the socket uses name for communicating with IDHOCNET
- Raw Sockets [33]-[34] for injecting the received IP based packet to the active IP based application, the packet is received by the standard TCP/UDP socket [35]-[36].

IDHOCNET uses ***monitor mode*** interface[37] for receiving and injecting the packets. The monitor mode allows injection of user defined packets and capturing of raw Wifi packets. The interface is a standard option and all present day Wifi cards support monitor mode.

4.2 Private IP Address Map

For supporting IP based applications in IDHOCNET a virtual interface say eth0, is used by IDHOCNET system. Whenever a node needs IP based application execution with other node a private IP address is initialized by using the ifconfig utility. A counter is maintained by the node which accounts for the number of private IP addresses already assigned for the peers in contact. As all the IP addresses are assigned and used local or private to the node therefore there is no need to maintain network wide uniqueness of the IP addresses. Therefore Auto configuration service is not required. Figure 15 shows a private IP address map of IDHOCNET when two nodes are in contact.



```
shahrukh@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1d:72:6b:2d:8e
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:17

eth0:1    Link encap:Ethernet  HWaddr 00:1d:72:6b:2d:8e
          inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          Interrupt:17

eth0:2    Link encap:Ethernet  HWaddr 00:1d:72:6b:2d:8e
          inet addr:10.0.0.3  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          Interrupt:17
```

Figure 15. Private IP address map of IDHOCNET

4.3 Threads of IDHOCNET

In order to efficiently perform various events, IDHOCNET main process is designed as a multithreaded IDHOCNET uses three in number POSIX [38][39] threads which perform the following functions:

- *Main thread* - receives wireless packets
- *IDServer thread* - receives IDbased application data
- *Netfilter thread* – receives IP packets data

4.4 ID Based Application Implementation

ID based application interaction is shown in Figure 16. At step 1 an ID based application creates its namespace socket with the name **/tmp/mysocket**. Names of different ID based applications are distinct. At step 2 IDHOCNET also creates its name socket with the name **/tmp/idserver**. A dedicated POSIX thread is used for processing the data captured by the IDHOCNET server socket. All the ID based applications will be required to connect to this server socket for sending their data to the IDHOCNET system. At step 3 the ID based application sends the data to the IDHOCNET socket. At step 4 the IDHOCNET receives the data. In step 5 IDHOCNET sends the received data to the ID based application. In step 6 the application consumes the data.

5. IDHOCNET Process Flows

5.1 Communication Establishment Mechanism

In order to exchange application data a communication path between the peers is required to be established. Reactive and proactive approaches are two popular path establishment mechanisms in ad hoc networking. The architecture of IDHOCNET supports both types of approaches. Presently, the implemented mode of path establishment strategy is adapted from simplified version of AODV[40] protocol known as AODVjr [41]. Functioning of AODV protocol is available in [42]. A number of additional procedures are added in the protocol for supporting additional functionalities supported by IDHOCNET. Moreover the biggest difference between the traditional ad hoc networking path establishment strategy is based on IP address and TCP/IP stack, whereas IDHOCNET is based on real identifiers. In the ensuing paragraphs a detail of path establishment mechanism is presented.

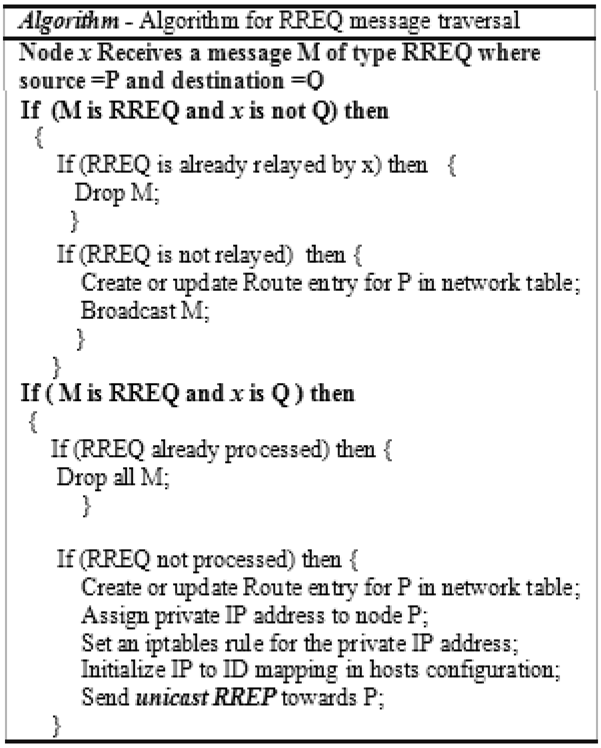
The path establishment process is started, whenever a node wants to send data to another node. All nodes in IDHOCNET support Route Request and Route Reply procedure as shown in the algorithms at Figure 17 and Figure 18 respectively.

As an example shown in Figure 19, Node A with identifier ‘123’ wants to establish communication with another node D with identifier ‘126’. At step 1, node A starts an ID based application and initiates a Route Request (RREQ) toward node D. At step 2 IDHOCNET transmits the RREQ packet through the Wireless interface operating in monitor mode. Now the RREQ packet traversed through the intermediate nodes as shown in Figure 20. During each step of the traversal a reverse path towards node A is initialized in each node which processes the RREQ. When the packet reaches the node D, which is the destination node a number of actions occur which include:

- Private IP address allocation for node A *(20.0.0.2)*
- IP table rule for outgoing packets for the respective private IP address
- An IP versus ID entry in the host configuration file

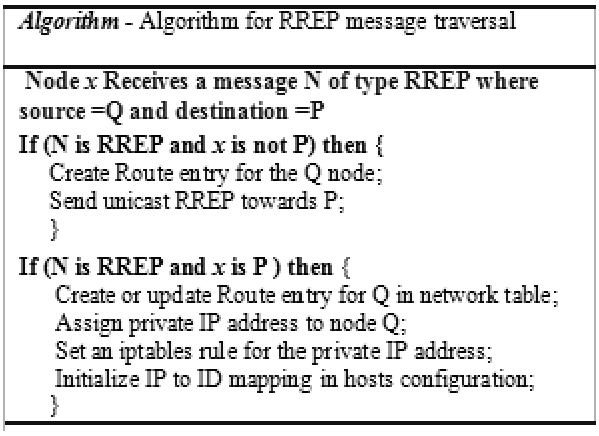
IPtable rule for the outgoing packet is added to capture the packet with destination address 20.0.0.2 in the IDHOCNET core application and make it compatible to ID based packet for further transmission. The last action of mapping in the host configuration files, allows running all IP based applications as ID based applications.

Now node D initiates a RREP message towards node A. During the traversal each node sends unicast RREP as per the already configured path. Finally the RREP packet reaches node A at step 3, as shown in Figure 19. At step 4 a private IP address for node D is initialized. At step 5 an IPtable rule for outgoing packet is added for node D’s private IP address. At step 6 an IP versus ID mapping is added to the host configuration file. At step 7 the IDHOCNET sends the connected message to the ID based application.



```
Algorithm - Algorithm for RREQ message traversal
Node x Receives a message M of type RREQ where
source =P and destination =Q
If (M is RREQ and x is not Q) then
  {
    If (RREQ is already relayed by x) then {
      Drop M;
     }
    If (RREQ is not relayed) then {
      Create or update Route entry for P in network table;
      Broadcast M;
      }
  }
If ( M is RREQ and x is Q ) then
 {
   If (RREQ already processed) then {
    Drop all M;
      }

    If (RREQ not processed) then {
      Create or update Route entry for P in network table;
      Assign private IP address to node P;
      Set an iptables rule for the private IP address;
      Initialize IP to ID mapping in hosts configuration;
      Send unicast RREP towards P;
  }
```

Figure 17. RREQ processing algorithm



```
Algorithm - Algorithm for RREP message traversal

Node x Receives a message N of type RREP where
source =Q and destination =P
If (N is RREP and x is not P) then {
   Create Route entry for the Q node;
   Send unicast RREP towards P;
   }
If (N is RREP and x is P ) then {
    Create or update Route entry for Q in network table;
    Assign private IP address to node Q;
    Set an iptables rule for the private IP address;
    Initialize IP to ID mapping in hosts configuration;
   }
```

Figure 18. RREP processing algorithm

5.2 Application execution process flows

5.2.1 IP application process flow

After the path has been established between the end points it is possible to run conventional IP based or ID based applications between the peers. Due to the presence of IP to ID binding in the hosts configuration file a user can directly use the identifier of the peer to run conventional IP based applications. The IP packet generated is captured by the IDHOCNET by making of Libnetfilter_queue and Libnfnetlink [43]. Further the IP address is used for resolving the identifier associated with the private IP address and the constructed ID based packet is sent through the packet socket interface. Figure 21 and 22 show the process flow of IP based application between node A and node D.

In Figure 21 at step1 user A issues PING command with the identifier of node D. The private IP address of node D is resolved by gethostbyname() call embedded in the PING application at step 2. After the resolution of the IP address at step 3 the IP address is available in the PING application. The application socket sends the data to the protocol stack at step 4. Due to the IPtable rule for the outgoing packet IDHOCNET captures the IP packet at step 5. The next hop identifier for node D is resolved at step 6 and IDHOCNET transmits the packet for Node D.

The packet is received after the routing by intermediate nodes by Node D. In Figure 22 the wireless interface of Node D receives the packet at step 1. The private IP address configured for node A is acquired from the network table. IDHOCNET injects the packet to the protocol stack in step 2. After the injection an outgoing PING reply is generated by the system. Due to the configured IPtable rule the IP packet is captured by IDHOCNET at step 3. The nexthop identifier is determined for node A at step 4 and packet is transmitted for routing by the intermediate nodes.

After the packet reaches Node A as shown in Figure 21, the private IP address for node D is found from the network table at step 7. IP packet is constructed and injected to the TCP/IP stack at step 8. The packet is received by the PING application at step 9.

5.2.2 *ID Application Process Flow*

ID based application is a novel paradigm and offer an alternative to IP based applications. Figure 23 and 24 shows an identifier based PING application. Throughout the process flow there is no requirement to refer IP address. Moreover TCP/IP protocol stack is not involved in ID based application provisioning. In Figure 23 at step 1 node A sends an ID based ping to node D by using an ID based application. The data is received by IDHOCNET at step 2. Network table is referred to see the next hop address at step 3. ID based packet is sent through the wireless interface at step 4. The packet is traversed through the intermediate nodes and reaches node D. As shown in Figure 24, node D receives the packet at step 1. The network table is referred at step 2 to know the next hop identifier for node A. The packet is analyzed by IDHOCNET as ID based ping request. An ID based ping reply is sent at step 2. The ID based ping reply reaches back to node A as shown in Figure 23 at step 5. IDHOCNET sends the ping reply to the respective application step 6.

6. Experimentation and evaluations

6.1 Experimental settings

An experimental setup was prepared to run various experiments. On each node Ubuntu 12.04 LTS[44] was install as operating system. In order to provide exactly similar protocol stack the operating system was installed using an image built by Remastersys [45]. Each node supported three types of operating and forwarding mechanisms or protocols which include:

- IDHOCNET – Identifier based forwarding
- OLSR - Layer3 (IP based forwarding)
- 802.11s – Layer2 (Link Layer based forwarding)

OLSR and 802.11s were used for comparing the data flows of IDHOCNET with standard IP based environments. In total 06 in number machines were used to build a multihop testbed as shown in Figure 25. Number of hops were increased one by one and data was send by using one of the forwarding mechanism as discussed above. Packet data was collected using Wireshark tool [46] for further analysis.

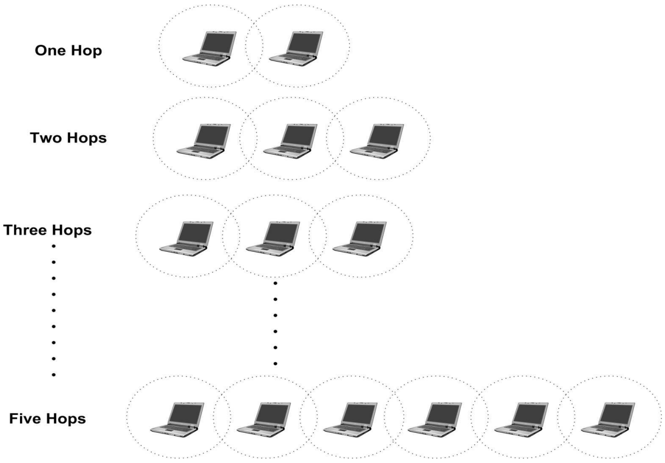


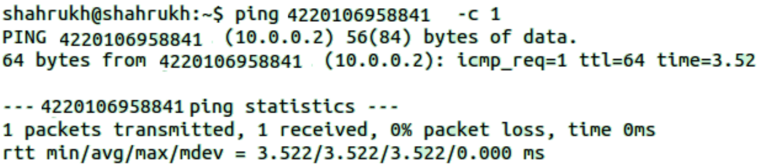
Figure 25. Multihop testbed

6.2 IP based applications execution

Various IP based application were executed on the IDHOCNET platform. Due to the feature of directly using the node identifier of the remote peer it is possible to run all the IP based applications using the identifier.

6.2.1 *Ping*

IDHOCNET was used for running an IP based PING application. Identifiers ware configured on the host and PING command was issued after the path establishment between the peers. Figure 26 shows the successful PING operation. It can be seen that identifier is used for running the PING application.



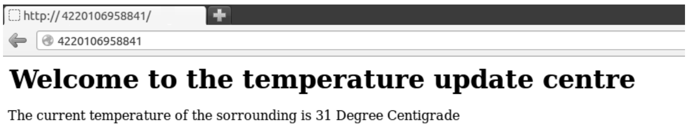
```
shahrukh@shahrukh:~$ ping 4220106958841   -c 1
PING 4220106958841  (10.0.0.2) 56(84) bytes of data.
64 bytes from 4220106958841  (10.0.0.2): icmp_req=1 ttl=64 time=3.52

--- 4220106958841 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.522/3.522/3.522/0.000 ms
```

Figure 26. Ping application using ID of the peer

6.2.2 *Hosted Web Application*

In another configuration setting a Client Server based scenario was tested using IDHOCNET. In one of the IDHOCNET node with identifier 4220106958841 an APACHE HTTP [47] server was installed. An application with functionality to tell the current temperature of the surrounding was hosted on the server.

Different clients connected with the host. An application run from one of the client browser is depicted in Figure 27. The server service is requested using its identifier and the default application sent the current temperature.



Welcome to the temperature update centre

The current temperature of the sorrounding is 31 Degree Centigrade

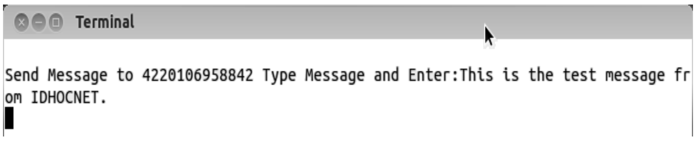
Figure 27. Accessing a hosted application by using browser

6.3 ID based applications execution

Novel ID based applications were implemented on the IDHOCNET nodes. Such applications do not require the use of IP based protocol stack. It can be seen that IP equivalent ID based applications can be designed.

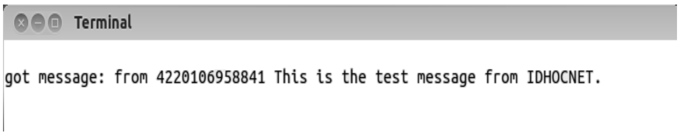
6.3.1 ID based chat application

An ID based chat application has been successfully developed as per the proposed design. After the path establishment phase a peer can send the data to the other end as shown in Figure 28. The sent data is received at the other end as shown in Figure 29.



```
Send Message to 4220106958842 Type Message and Enter:This is the test message fr
om IDHOCNET.
```

Figure 28. Sending data to multihop peer using ID based application

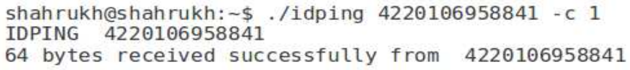


```
got message: from 4220106958841 This is the test message from IDHOCNET.
```

Figure 29. Receiving data from the remote peer

6.3.2 ID based PING application

An IP based equivalent application for PING was designed and implemented with the name IDPING. The application execution if shown in Figure 30. A ping request was sent for the respective identifier. After the ping reply has been received status of the received bytes is visible to the user.



```
shahrukh@shahrukh:~$ ./idping 4220106958841 -c 1
IDPING  4220106958841
64 bytes received successfully from  4220106958841
```

Figure 30. IDPing application execution

6.4 Overhead comparison for IP based applications

Table 2 shows the size of frames in Bytes required for forwarding a 20 byte IP packet with 30 bytes of payload. Figure 31 shows the graphical plot of the same data. The size of the frames used by IDHOCNET requires least number of bytes to transmit the required IP packet. Moreover it is also possible to not send the source and destination IPv4 addresses (frame at d) to save another 8 bytes. This option is possible due to the implementation of private IP addresses.

Table 2. Comparison of protocol overhead

Mechanism	Types	MAC (Bytes)	IP (Bytes)	Data (Bytes)	Total
a. Link Layer		64	20	30	114
b. IP Layer		52	20	30	102
c. IDHOCNET		46	20	30	96
d. IDHOCNET (No src,des IP)		46	12	30	88

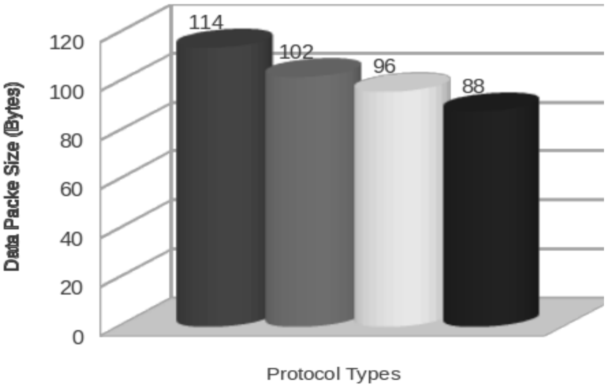


Figure 31. Size of different forwarding mechanisms

6.5 Analysis of VoIP Application

Voice over IP application are very popular in Internet. The core frame structure of VoIP application is composite and comprises of RTP, UDP and IP frames. An experiment to analyze the behaviour of RTP/UDP/IP was conducted on the multihop testbed. Three types of forwarding mechanisms were used which include IDHOCNET, OLSR and 802.11s. For each configuration number of hops was increased one by one and data was gathered by using Wireshark tool. The RTP/UDP/IP stream was generated by using RTPTools[48]. As shown In Figure 32 the delay jitter of the IDHOCNET is found to be lower than OLSR and 802.11s. The major factors which affect delay jitter include routing table updates, packet processing at the forwarding nodes and congestion in the network.

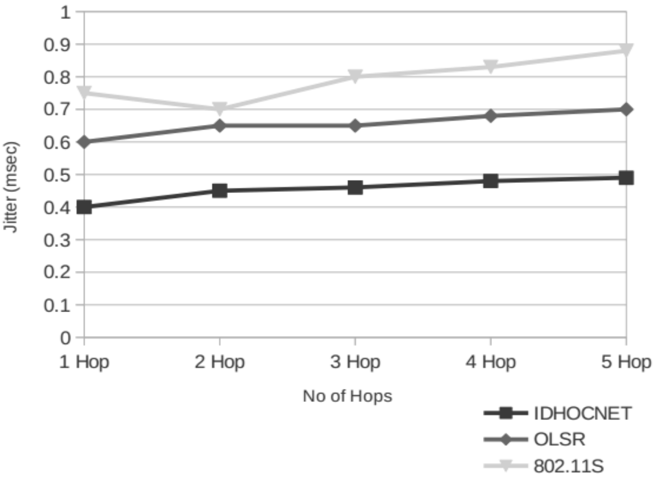


Figure 32. Delay Jitter in different forwarding mechanisms

Packet Loss ratio is also calculated for all the forwarding mechanisms in different hop configuration as shown in Figure 33. The packet loss ratio of 802.11s is more than OLSR and IDHOCNET mechanisms. The packet loss ratio of IDHOCNET and OLSR is almost similar.

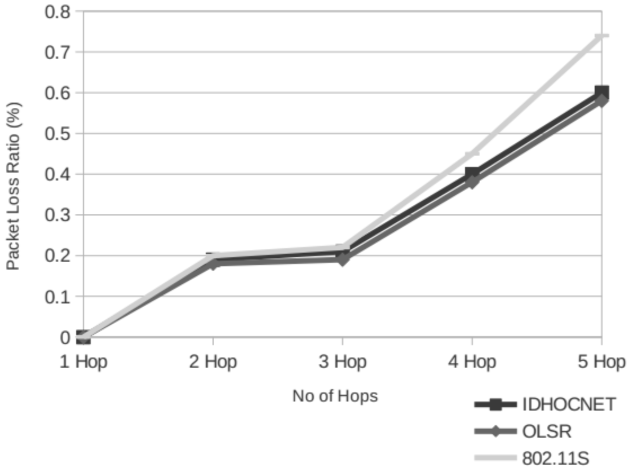


Figure 33. Packet Loss Ration in different mechanisms

7. Conclusions

An ID based Ad hoc Network (IDHOCNET) framework has been designed as an aim to solve number of limitations of IP based ad hoc networks. A prototype of the proposed system has been implemented in Linux platform. A novel ID based application paradigm has been implemented, where as contemporary IP based application can also be executed in the proposed system. In the proposed framework, it is possible to use identifier while running the IP based application. Future work may include development of a Kernel module for the implemented features of IDHOCNET which is expected to improve the system performance. Security provision is an important aspect for ad hoc networks future work may include security implementation for IDHOCNET framework.

References

[1] T. Tronco, “A Brief History of the Internet”, New Network Architectures, Springer, pp. 1–11, 2010

[2] E. Gerich, “Guidelines for management of IP Address Space”, RFC 1466, 1993.

[3] P. Mockapetris and K. J. Dunlap, “Development of the domain name system”, ACM SIGCOMM proceedings on Communications architectures and protocols, Vol. 18, No. 4, pp. 123-1337, 1988.

[4] R. Droms, “Dynamic host configuration protocol”, RFC 1541, 1997.

[5] P. Srisuresh and M. Holdrege, “IP network address translator (NAT) terminology and considerations”, RFC 2663, 1999.

[6] C. So-in, R. Jain, S. Paul, and J. Pan, “Virtualization architecture using the ID / Locator split concept for Future Wireless Networks (FWNs)”, Elsevier Journal of Computer Networks, Vol. 55, No. 2, pp. 415–430, 2011.

[7] R. Jain, “Internet 3.0: Ten problems with current internet architecture and solutions for the next generation”, IEEE Military Communications Conference, pp. 1–9, 2006

[8] H. Zhou and M. W. Mutka, “Review of Autoconfiguration for MANETs”,Wireless Ad-hoc Networks, Intech, pp. 123–144, 2012

[9] L. J. García Villalba, J. Garcia Matesanz, A. L. Sandoval Orozco, and J. D. Marquez Díaz, “Auto-configuration protocols in mobile ad hoc networks”, Sensors, Vol. 11, No. 4, pp. 3652–3666, 2011.

[10] M. Masdari, M. Maleknasab, and M. Bidaki, “A survey and taxonomy of name systems in mobile ad hoc networks,” Elserview Journal of Network and Computer Applications, Vol. 35, No. 5, pp. 1493–1507, 2012.

[11] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, “Internet indirection infrastructure”, IEEE/ACM Transactions on Networking, Vol. 12, No. 2, pp. 205–218, 2004.

[12] C. So-in, R. Jain, S. Paul, and J. Pan, “Future Wireless Networks : Key Issues and a Survey (ID / Locator Split Perspective),” International Journal of Communication Networks and Distributed Systems, Vol. 8, No. 1, pp. 24–52, 2012.

[13] W. Ramirez, X. Masip-Bruin, M. Yannuzzi, R. Serral-Gracia, A. Martinez, and M. S. Siddiqui, “A survey and taxonomy of ID/Locator Split Architectures,” Elsevier Journal of Computer Networks, Vol. 60, pp. 13–33, 2014.

[14] V. P. Kafle and M. Inoue, “HIMALIS: Heterogeneity inclusion and mobility adaptation through locator id separation in new generation network,” IEICE Transactions on Communications, vol. 93, no. 3, pp. 478–489, 2010.

[15] J. Kim, H. Jung, and S. Koh, “Mobile Oriented Future Internet (MOFI): Architectural Design and Implementations”, ETRI Journal, Vol. 35, No. 4, pp. 666–676, 2013.

[16] J. Pan, S. Paul, R. Jain, and M. Bowman, “MILSA : A Mobility and Multihoming Supporting Identifier Locator Split Architecture for Naming in the Next Generation Internet”, IEEE GLOBECOM, pp. 1-6, 2008.

[17] J. Pan, R. Jain, S. Paul, M. Bowman, X. Xu, and S. Chen, “Enhanced MILSA Architecture for Naming, Addressing, Routing and Security Issues in the Next Generation Internet,” IEEE Communication Conference, pp. 1–6, 2009

[18] B.-O. Kwak, T.-H. Lee, and W. Chun, “ID based communication in Domain-Insulated Autonomous Network Architecture (DIANA),” IEEE ICT Convergence Conference, pp. 264–269, 2012

[19] D. Rodr, E. Munthe-kaas, and T. Plagemann, “ILORIN : Identifier-LOcator Resolution for Infrastructure-less Networks,” IEEE Network of the Future (NOF) Third International Conference, pp. 1–8, 2012

[20] W. Haiquan, C. Meng, H. Junshun, and X. Chunhe, “Scalable and Identifier/Locator-Splitting Routing Protocol for Mobile Ad Hoc Networks”, China Communications, Vol. 9, No. 1, pp. 102–110, 2012.

[21] T. Reshmi and K. Murugan, “Filter-based address autoconfiguration protocol (FAACP) for duplicate address detection and recovery in MANETs”, Springer Computing, 2014.

[22] S. Huq and S. Begum, “Ensemble Approach for IP Auto-configuration in Ad Hoc MANETs,” Springer Mobile Communication and Power Engineering, pp. 193–199, 2013.

[23] W. Xiaonan, Y. Yuan, Y. Yufeng, and C. Hongbin, “An address configuration protocol for 6LoWPAN wireless sensor networks based on PDAD,” Computer Standard and Interfaces, No. 6, pp. 918–927, 2014.

[24] S. Singh, N. Rajpal, and A. Sharma, “Address allocation for MANET merge and partition using cluster based routing,” Springerplus, Vol. 1, 2014.

[25] J. Rodríguez-Covili, S. F. Ochoa, and J. a. Pino, “High level MANET protocol: Enhancing the communication support for mobile collaborative work”, Elsevier Journal of Network and Computer Applications, Vol. 35, No. 1, pp. 145–155, 2012.

[26] J. RodríGuez-Covili and S. Ochoa, “A communication infrastructure to ease the development of mobile collaborative applications”, Elsevier Journal of Network and Computer Applications , Vol. 6, No. 34, pp. 1883–1893, 2011.

[27] S. Khalid and A. Mahboob, “Design and Implementation of ID based MANET Auto-configuration Protocol,” International Journal of Communication Networks and Information Security, Vol. 5, No. 3, pp. 141–151, 2013.

[28] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot, and others, “Optimized link state routing protocol (OLSR)”, IEEE INMIC, 2001.

[29] R. C. Carrano, L. C. S. Magalhaes, D. C. M. Saade, and C. V. N. Albuquerque, “IEEE 802.11s Multihop MAC: A Tutorial,” IEEE Communication Surveys & Tutorials, Vol. 13, No. 1, pp. 52–67, 2011.

[30] P. Neira‐Ayuso, “Communicating between the kernel and user‐space in Linux using Netlink sockets,” Wiley Journal of Software Practices & Experience, Vol. 40, No. 9, pp. 797–810, 2010.

[31] M. Barbeau and E. Kranakis, “Wireless Network Programming,” Principles of Ad Hoc Networking, John Wiley & Sons, pp. 103–111, 2007

[32] C. LLC, M. L. Mitchell, A. Samuel, and J. Oldham, Advanced Linux Programming, Sams Publishing, pp. 120–123, 2001

[33] C. Nagel, A. Mungale, and V. Kumar, “Raw Socket Programming”, Pro .NET 1.1 Network Programming, Springer, 2004, pp. 163–187.

[34] R. Rosen, “Internet control message protocol (ICMP)”, Linux Kernel Networking, Apress, pp. 37–61, 2014.

[35] B. Shafiei and A. Branch, “Review Socket Programming,” Aust. Journal of Basic & Applied Sciences, Vol. 6, No. 7, pp. 354–364, 2012.

[36] A. Tudzarov and T. Janevski, “Design for 5G mobile network architecture,” International Journal of Communication Networks and Information Security, Vol. 3, No. 2, pp. 112–123, 2011.

[37] M. Vipin, “Analysis of Open Source Drivers for IEEE 802 . 11 WLANs”, IEEE ICWCSC, pp. 1–5, 2010.

[38] U. Drepper and I. Molnar, “The Native POSIX Thread Library for Linux,” URL http//people. redhat. com/drepper/nptl-design. pdf, 2007.

[39] U. Drepper and I. Molnar, “The native POSIX thread library for Linux,” White Pap. Red Hat Inc, 2003.

[40] I. D. Chakeres and E. M. Belding-royer, “AODV Routing Protocol Implementation Design,” IEEE Distributed Computing Systems Workshops, pp. 698–703, 2004

[41] I. D. Chakeres and L. Klein-berndt, “AODVjr , AODV Simplified”, ACM SIGMOBILE Mob. Comput. Commun. Rev, Vol. 6, No. 3, pp. 100–101, 2002.

[42] H. Zafar, N. Alhamahmy, D. Harle, and I. Andonovic, “Survey of Reactive and Hybrid Routing Protocols for Mobile Ad Hoc Networks,” International Journal of Communication Networks and Information Security, Vol. 3, No. 3, pp. 193–216, 2011.

[43] “netfilter/iptables project homepage - The netfilter.org ‘libnfnetlink’ project.”[Online]. Available: http://www.netfilter.org/projects/libnfnetlink/. [Accessed: 29-Jan-2015].

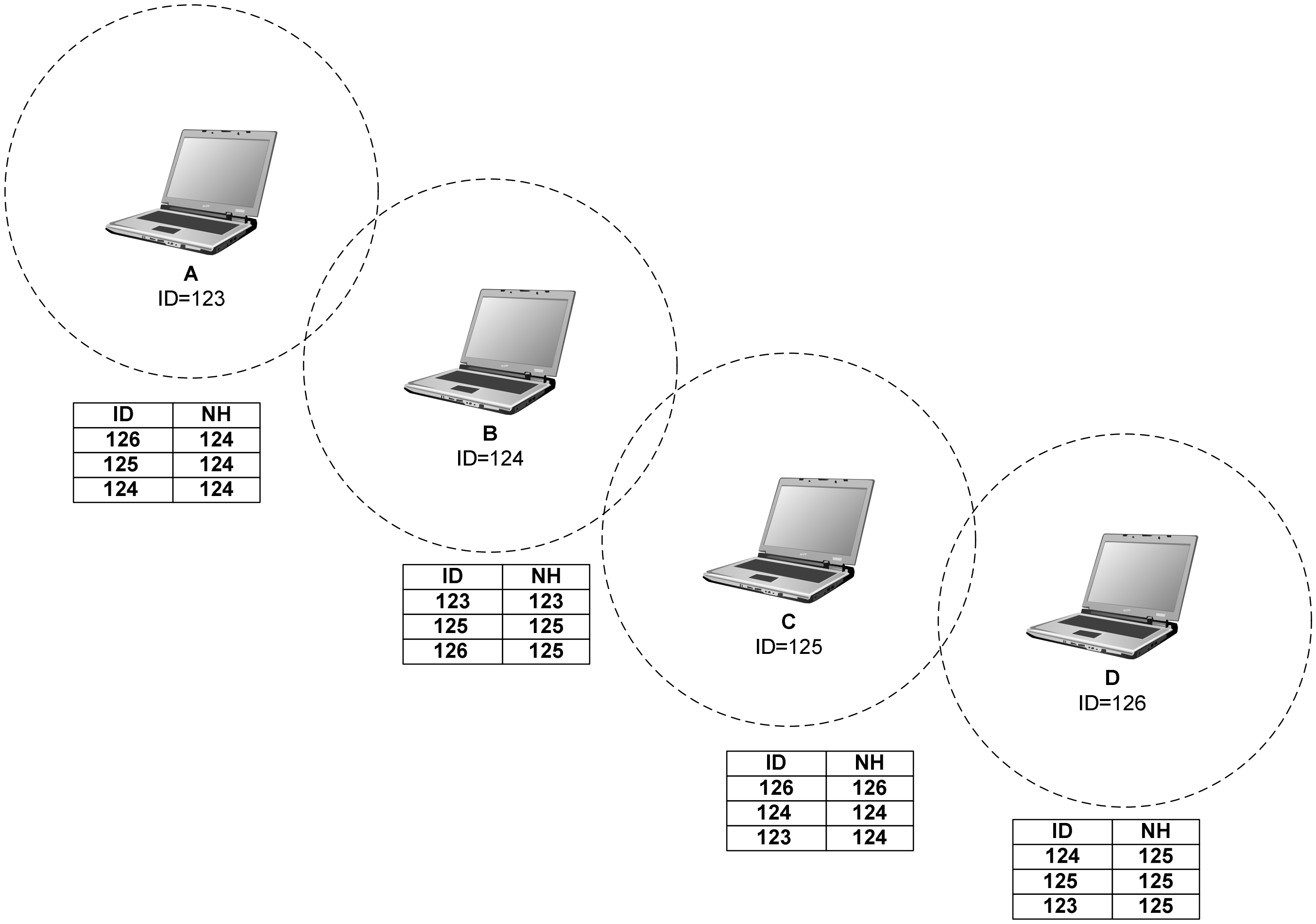
[44] K. Thomas, J. Sicam, D. James, and E. Hewitt, “Beginning Ubuntu Linux”,Apress, 2007.

[45] T. Brijeski, “Remastersys.” [Online]. Available: http://www.remastersys.com/. [Accessed: 29-Jan-2015].

[46] R. Shimonski, “The Wireshark Field Guide: Analyzing and Troubleshooting Network Traffic”, Newnes, 2013.

[47] B. Laurie and P. Laurie, “Apache: The definitive guide”, O’Reilly Media, Inc., 2003.

[48] H. Schulzrinne, “RTP Tools 1.20.” [Online]. Available: http://www.cs.columbia.edu/irt/software/rtptools/. [Accessed: 29-Jan-2015].



ID	NH
126	124
125	124
124	124

ID	NH
123	123
125	125
126	125

ID	NH
126	126
124	124
123	124

ID	NH
124	125
125	125
123	125

Figure 4. Routing and Identification on the basis of identifiers

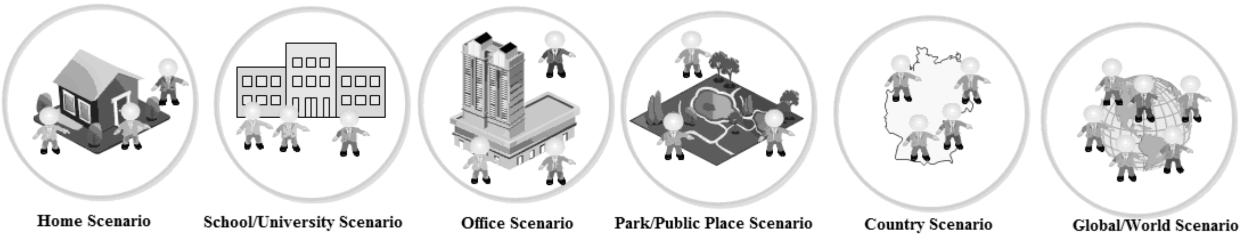


Figure 7. Different realms for ad hoc network settings

Table 1. Comparison of IP based ad hoc networks and IDHOCNET

Aspect	IP Centric	ID Centric
End point Identification	Uses IP addresses	Uses real world identifiers
Auto-configuration	Requires Auto-configuration service	Not Required
Realm support	No concept of realm exists	Supports formation of different realms
Network configuration	SSID, Auto configuration, IP address, IP Address particular series , Naming and name resolution	Minimal only pre configured identifiers are required
Identifiers in a packet	MAC Address and IP addresses	Only identifiers are transmitted
Routing basis	IP addresses	Identifiers
Naming & Name Resolution	Requires naming and name resolution	Not required
Multi-homing	Does not support	Supports multihoming
Mobility Issue	Complex to handle	Simple to handle
Backward Compatibility	Supports only IP based services	Supports ID and IP services
Uniformity	Different nodes assume roles to support mechanisms like Auto configuration, name resolution, etc.	All nodes are truly peers and assume similar role.
Complexity	Complex software stack due to multiple configuration and service requirements	Simple identifier based software stack
IP address variation	Variations are expected due to dynamic environment	No variations due to private address mapping
Connectivity support to ID/Loc split architectures	Not adaptable	Easily adaptable
Application Support	IP based applications	IP based applications and novel ID based applications

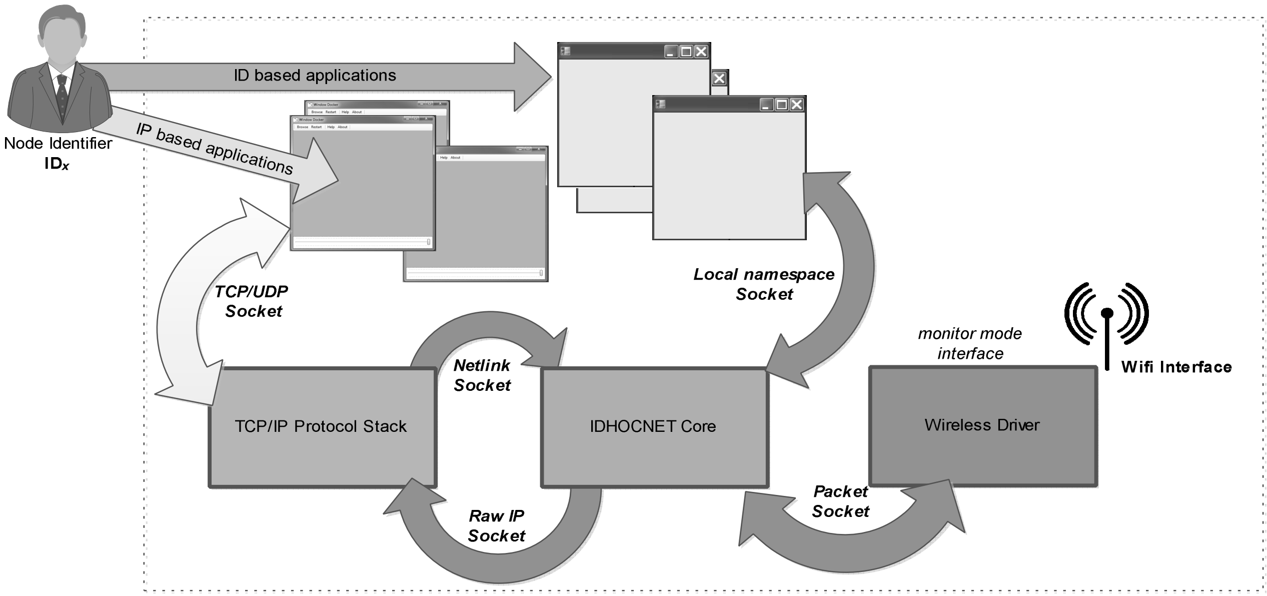
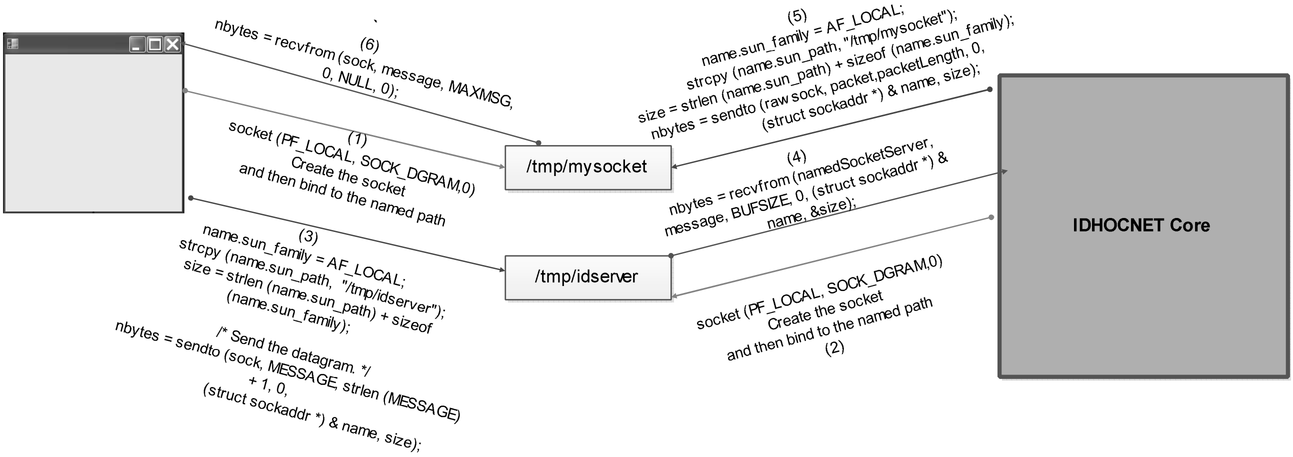


Figure 14. Sockets API interactions of IDHOCNET



```
nbytes = recvfrom (sock, message, MAXMSG,
0, NULL, 0);
```

```
socket (PF_LOCAL, SOCK_DGRAM,0)
Create the socket
and then bind to the named path
```

```
name.sun_family = AF_LOCAL;
strcpy (name.sun_path, "/tmp/idserver");
size = strlen (name.sun_path) + sizeof
(name.sun_family);

/* Send the datagram. */
nbytes = sendto (sock, MESSAGE, strlen (MESSAGE)
+ 1, 0,
(struct sockaddr *) & name, size);
```

```
name.sun_family = AF_LOCAL;
strcpy (name.sun_path, "/tmp/mysocket");
size = strlen (name.sun_path) + sizeof (name.sun_family);
nbytes = sendto (raw sock, packet,packetLength, 0,
(struct sockaddr *) & name, size);
```

```
nbytes = recvfrom (namedSocketServer,
message, BUFSIZE, 0, (struct sockaddr *) &
name, &size);
```

```
socket (PF_LOCAL, SOCK_DGRAM,0)
Create the socket
and then bind to the named path
```

Figure 16. ID based application design and interaction in IDHOCNET

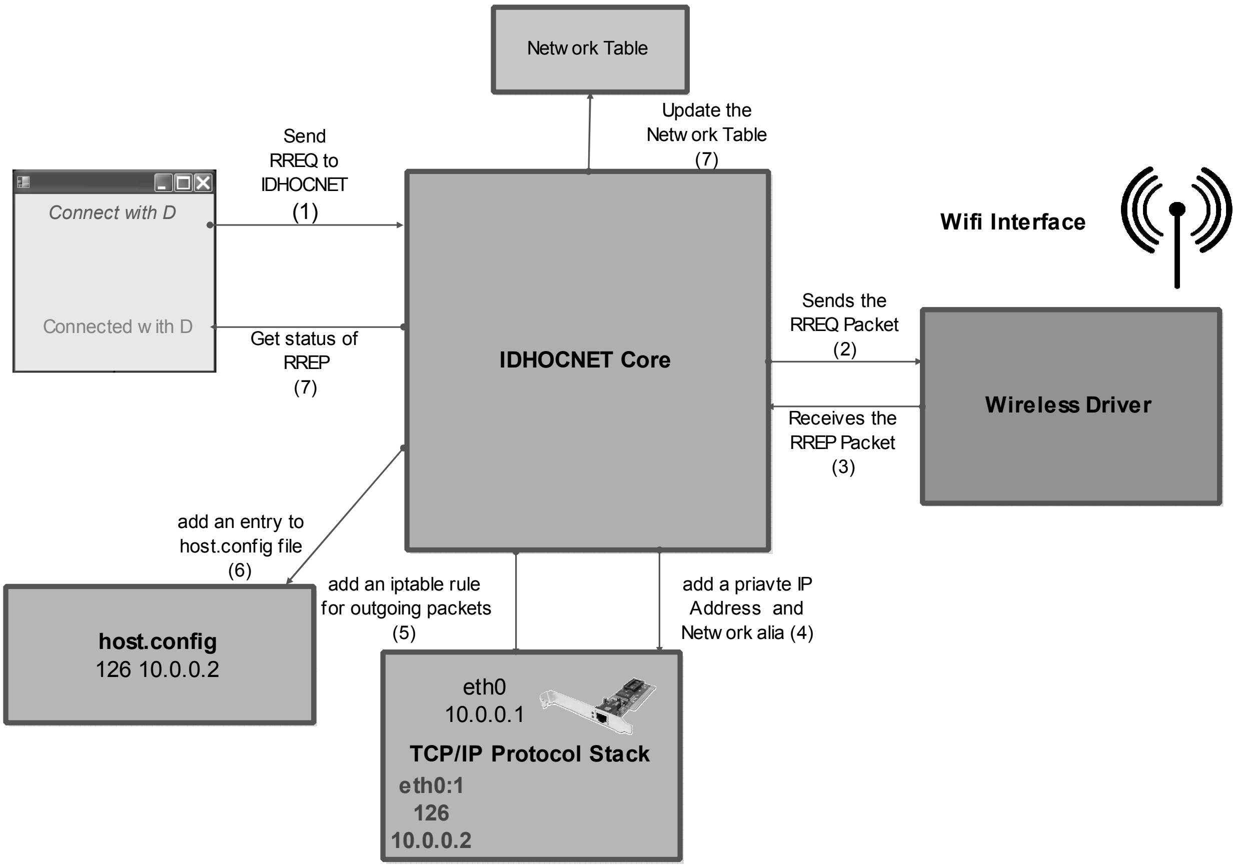
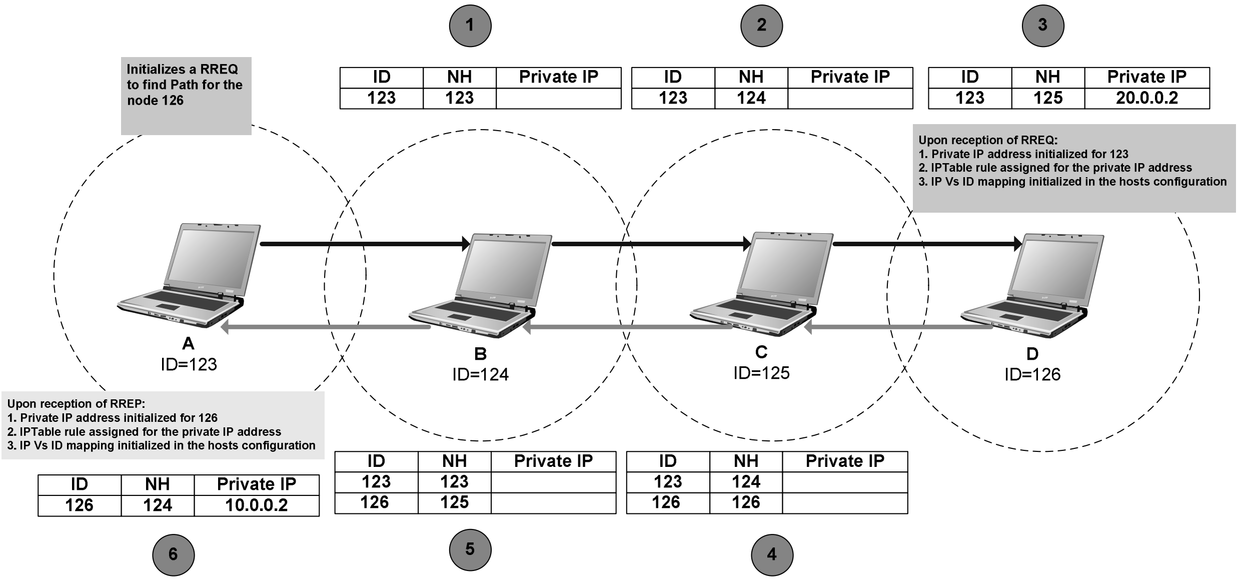


Figure 19. Communication establishment between peers


Initializes a RREQ to find Path for the node 126
1
2
3
Upon reception of RREQ:
1. Private IP address initialized for 123
2. IPTable rule assigned for the private IP address
3. IP Vs ID mapping initialized in the hosts configuration
A
ID=123
B
ID=124
C
ID=125
D
ID=126
Upon reception of RREP:
1. Private IP address initialized for 126
2. IPTable rule assigned for the private IP address
3. IP Vs ID mapping initialized in the hosts configuration
6
5
4

1

ID	NH	Private IP
123	123	

2

ID	NH	Private IP
123	124	

3

ID	NH	Private IP
123	125	20.0.0.2

6

ID	NH	Private IP
126	124	10.0.0.2

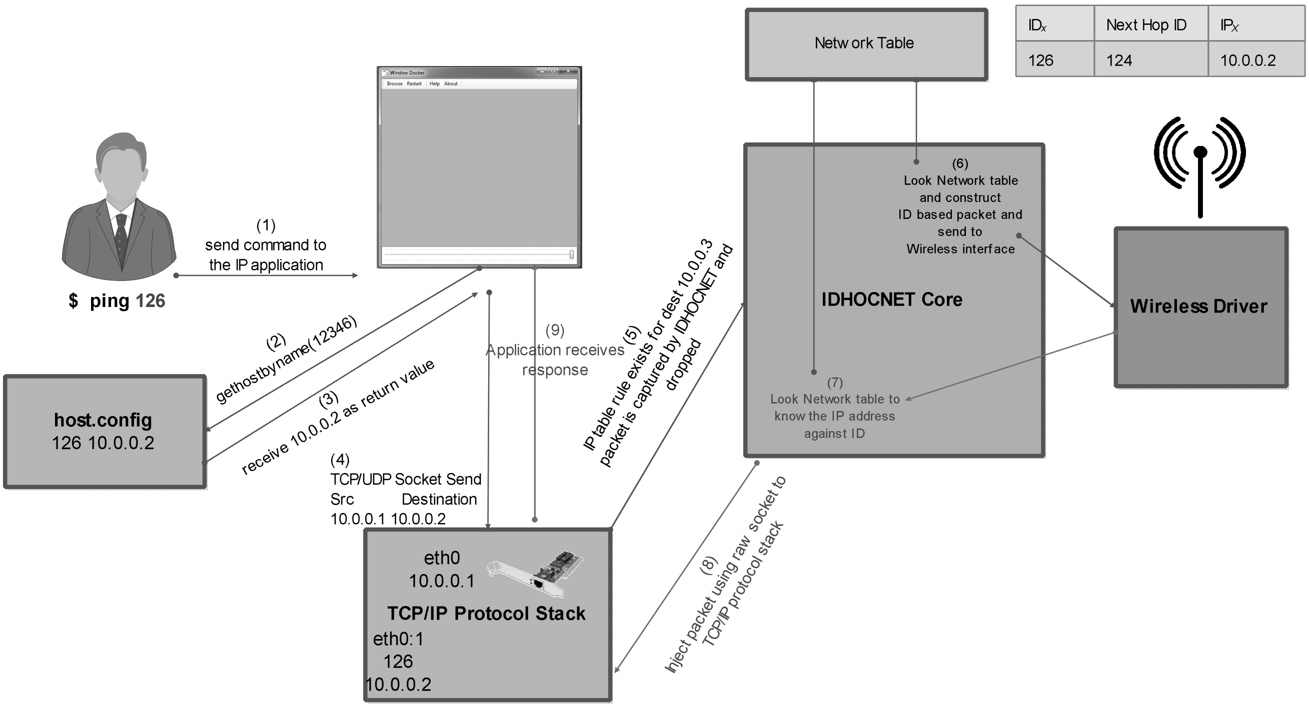
5

ID	NH	Private IP
123	123	
126	125	

4

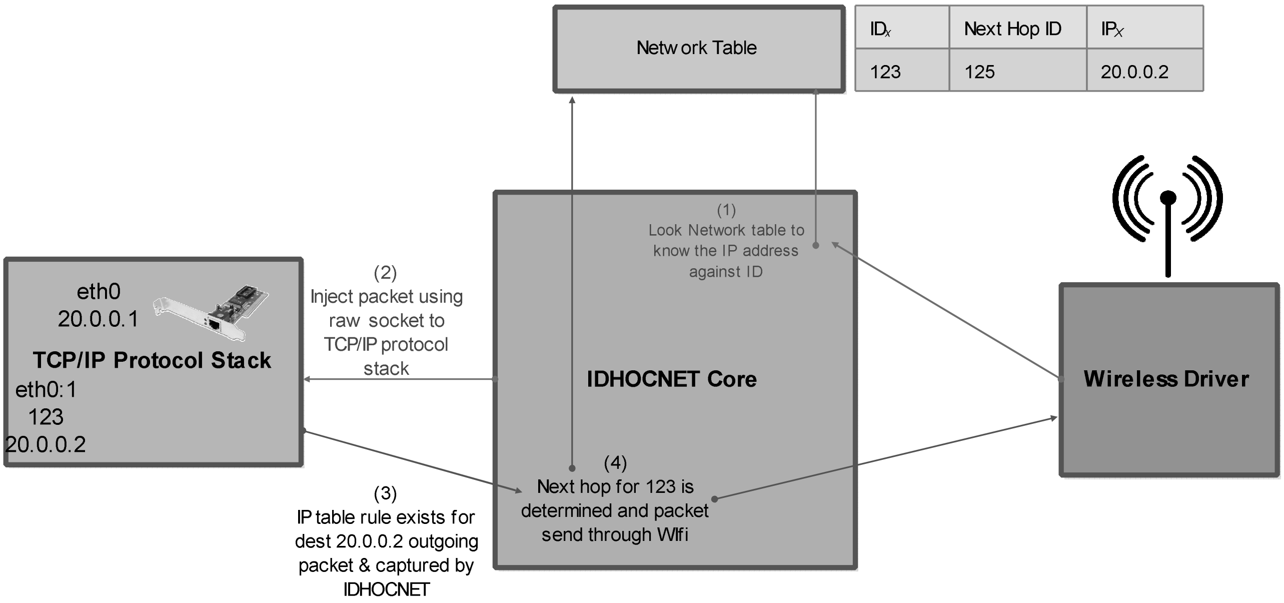
ID	NH	Private IP
123	124	
126	126	

Figure 20. Flow of RREQ and RREP between IDHOCNET nodes



ID_x	Next Hop ID	IP_X
126	124	10.0.0.2

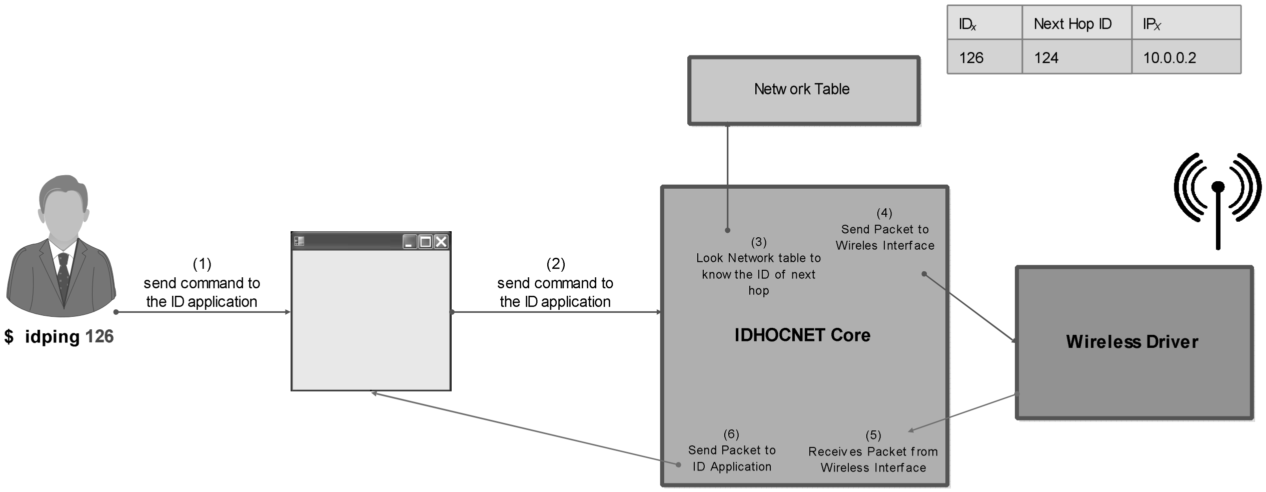
Figure 21. Data exchange between IDHOCNET and IP based application



IDX	Next Hop ID	IPX
123	125	20.0.0.2

Netw ork Table
eth0
20.0.0.1
TCP/IP Protocol Stack
eth0:1
123
20.0.0.2
(2)
Inject packet using raw socket to TCP/IP protocol stack
(1)
Look Network table to know the IP address against ID
IDHOCNET Core
Wireless Driver
(4)
Next hop for 123 is determined and packet send through Wifi
(3)
IP table rule exists for dest 20.0.0.2 outgoing packet & captured by IDHOCNET

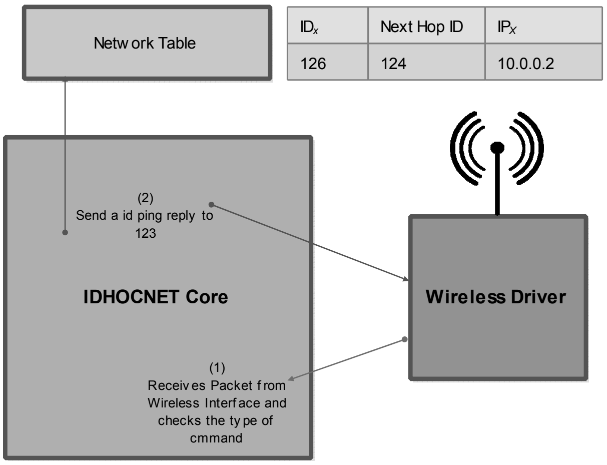
Figure 22. Remote peer view of IP based application



IDX	Next Hop ID	IPX
126	124	10.0.0.2

Netw ork Table
(1)
send command to the ID application
$ idping 126
(2)
send command to the ID application
(3)
Look Network table to know the ID of next hop
(4)
Send Packet to Wireles Interface
IDHOCNET Core
Wireless Driver
(6)
Send Packet to ID Application
(5)
Receiv es Packet from Wireless Interface

Figure 23. Data exchange between IDHOCNET and ID based application



IDX	Next Hop ID	IPX
126	124	10.0.0.2

Netw ork Table
(2)
Send a id ping reply to 123
IDHOCNET Core
Wireless Driver
(1)
Receiv es Packet from Wireless Interface and checks the ty pe of cmmand

Figure 24. Remote peer view for ID based application