Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2016, Article ID 6438584, 18 pages
http://dx.doi.org/10.1155/2016/6438584

Research Article

IDHOCNET: A Novel ID Centric Architecture for Ad Hoc Networks

Shahrukh Khalid,[1] Athar Mahboob,[2] Choudhry Fahad Azim,[3] and Aqeel Ur Rehman[4]

[1]*Telecommunication Engineering, Hamdard University, Sharae Madinat Al-Hikmah, Muhammad Bin Qasim Avenue, Karachi 74600, Pakistan*
[2]*Khawaja Fareed University of Engineering & Information Technology, Rahim Yar Khan 64200, Pakistan*
[3]*Faculty of Engineering Sciences and Technology, Hamdard University, Sharae Madinat Al-Hikmah, Muhammad Bin Qasim Avenue, Karachi 74600, Pakistan*
[4]*Department Computer Science, Hamdard University, Sharae Madinat Al-Hikmah, Muhammad Bin Qasim Avenue, Karachi 74600, Pakistan*

Correspondence should be addressed to Shahrukh Khalid; shahrukh_khalid@hotmail.com

Received 8 November 2015; Accepted 28 January 2016

Academic Editor: Rui Zhang

Ad hoc networks lack support of infrastructure and operate in a shared bandwidth wireless environment. Presently, such networks have been realized by various adaptations in Internet Protocol (IP) architecture which was developed for infrastructure oriented hierarchical networks. The IP architecture has its known problem and issues even in infrastructure settings, like IP address overloading, mobility, multihoming, and so forth. Therefore, when such architecture is implemented in ad hoc scenario the problems get multiplied. Due to this fact, ad hoc networks suffer from additional problems like IP address autoconfiguration, service provisioning, efficient bandwidth utilization, and node identification. In this paper we present IDHOCNET which is a novel implementation of service provisioning and application development framework in the ad hoc context. We illustrate a number of implemented features of the architecture which include IP address autoconfiguration, identification of nodes by using real world identifiers, IP based services support in ad hoc networks, and a new class of application known as ID based application. Moreover how identifiers can completely replace the IP addresses to run the IP based applications is shown. It is expected that this work will open new research horizons and paradigms for ad hoc networks.

1. Introduction

The current Internet is built upon Internet Protocol (IP) architecture [1]. It provides numerous services like Voice over Internet Protocol (VoIP), email, gaming, social networking, e-banking, and so forth. However, the classical IP architecture suffers a number of known issues highlighted in [2, 3] in next generation networks. Such problems include IP address overloaded semantics, multihoming, and mobility. In order to meet the demanding needs from the Internet, a number of proposals in the direction of new architectures have emerged. The most popular trend among the new architectures is the ID/Locator based Split Architecture (ILSA). ILSA based systems are derived from the seminal paper at [4]. A comprehensive taxonomy and a review of such architectures are given in [5].

Continuing with the same premise of problems associated with infrastructure based networks, when IP based architecture is applied in the scenario of ad hoc networks the problems tend to increase manifoldly due to the fact that such networks lack infrastructure support. Additionally, IP addresses have topological meaning but ad hoc networks have flat topology and may lack hierarchy. There are no centralized servers in ad hoc networks for resolving issues like IP address conflicts using DHCP (Dynamic Host Configuration Protocol) [6] or DHCPv6 [7]. Moreover, there can be no central approach for name resolution like the Domain Name System (DNS) [8]. As a result, there are numerous of proposals in

the direction of distributed naming and name resolution [9] for ad hoc networks.

IP address holds a pivotal presence in IP based architecture. For the functioning of a network it is essential that each communicating entity is configured with a unique IP address. Such a requirement is fulfilled with the help of centralized DHCP servers in infrastructure based networks. However ensuring unique IP address in ad hoc networks is a nontrivial issue. IP address conflict problem is studied under the field of IP address configuration services or IP address configuration protocols. In this pursuit various approaches have been proposed to resolve the problem of IP address autoconfiguration in ad hoc networks [10–12]. There are numerous papers suggesting such techniques but practical implementation of IP address conflict resolution schemes is hard to find. In an IP based ad hoc network a fully active IP address conflict scheme should be present to cater for the IP address conflict issues.

Bandwidth preservation is another important consideration in ad hoc networking scenario in which nodes have scarce resources and shared medium. Any measure which could save the number of bytes while transmitting the data is very valuable. A popular bandwidth preservation technique is adoption of an IP header compression scheme. In IP header compression a context or relationship is established within one hop peer. On the basis of the established relationship or context there is a possibility of compressing the IP header at the sender end and decompressing or recovering the complete packet at the receiver end. Thus a large amount of bytes can be saved and IP protocol overhead can be reduced. In ad hoc networking scenario the arrangement of nodes is in multihop fashion; therefore the IP header compression scheme is required to be modified for bandwidth preservation. In IP based networking, realization of a multihop header compression service is an open issue.

From the above discussion it can be analysed that IP based architecture has problems in infrastructure based network realization and such problems become more dominant in ad hoc networking scenarios. Moreover, it is observed that ID/Locator split architectures cannot be directly applied in ad hoc networks due to nonpresence of required infrastructure support. A depiction of ad hoc network node in IP based network, facing multitude problems, is shown in Figure 1.

As a step for resolving such problems for ad hoc networks we implemented a preliminary design for IP address autoconfiguration using identifiers [13]. In this paper, we considerably extend our earlier work and propose a novel ID centric networking architecture for ad hoc networks with the following design goals:

(i) Introduction of identifier based paradigm for communication entities.

(ii) An ID based application design.

(iii) Backward compatibility for IP based services and Internet provisioning.

(iv) Use of identifiers like MAC addresses, telephone numbers, and so forth for directly running IP based services.

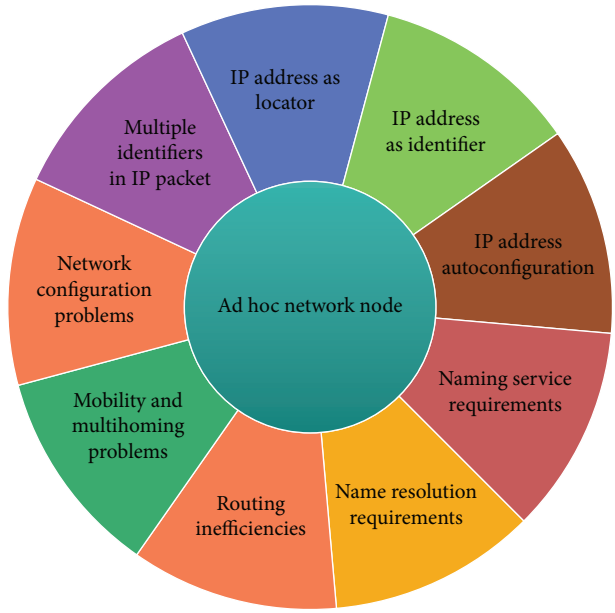


Figure 1: Problem circle of IP architecture based ad hoc networking.

(v) Multihop header compression when streaming services like VoIP are used.

In order to achieve the above-mentioned goals we implemented a functional prototype network which we call as identifier based ad hoc network (IDHOCNET). The proposed network system has been implemented on Linux based platform and a testbed has been built for conducting test and measurements. The remainder of our paper is organized as follows: Section 2 highlights the practical design considerations and problems, associated with IP based realization of ad hoc networks. Presence of such problems and issues has motivated us for design and development of our proposed ID centric architecture. Section 3 describes various components of our proposed architecture. Section 4 gives process flows of the architecture. Section 5 describes the implementation details of the prototype. Section 6 provides the experimental results. Section 7 compares the framework with respect to existing ad hoc schemes. Finally, Section 8 concludes the paper and presents future work.

2. Requirements, Issues, and Challenges

2.1. IP Address Autoconfiguration Service. When IP centric communication is used, it is mandatory for nodes to have unique IP addresses assigned in their interface software. For a large ad hoc network it is very difficult to assign IP addresses manually and ensure the uniqueness of IP addresses. IP address autoconfiguration is an open research field and the large number of proposed schemes exists. Although there are hundreds of proposals, practical implementation of any of such schemes is scarcely available. Most of the autoconfiguration schemes exist for mobile ad hoc network (MANET) due to dominant problems of network merging and partitioning. We give examples of few latest autoconfiguration protocols.

The protocol proposed in [14] uses proxy nodes for assigning IP addresses to the joining nodes. Each proxy node has disjoint set of IP address blocks. Each node has predistributed hash functions for assuring secure communication. The SAAMAN [15] is an IP address autoconfiguration mechanism in which routing is based on geoforwarding mechanism. In order to ensure uniqueness of IP addresses the protocol implements distributed servers known as Distributed Detection Servers (DDS). Extended Prime Number Allocation (EPNA) [16] uses proxy nodes for assigning unique IP addresses. The rule for assignment is based on disjoint IP address generation by prime factor multiple. The protocol in [17] detects duplicate IP address through OLSR routing messages. The same authors have extended their work for merging support in [18]. An IPV6 autoconfiguration scheme is proposed in [19] for Wireless Sensor Networks (WSN) which is based on location information of the sensor nodes. Dynamic WMN (Wireless Mesh Network) Configuration Protocol (DWCP) [20] is an autoconfiguration scheme for Wireless Mesh Networks. The algorithm exploits the hierarchal structure of the WMN. A tree based scheme is proposed in [21] in which root node performs diverse set of functions which includes maintenance of record in its database of all group leaders in the network, maintaining free address information held by group leaders, index of leader of leaders for assignment of new root in case of its own departure from the network, and also performing network merging and partitioning tasks. The group leaders contain disjoint block of IP addresses. Another tree based architecture is proposed in [22] in which IP addresses are assigned by proxy nodes. Control packet overhead is restricted to one-hop peers. The protocol proposed in [22] also supports address reclamation.

Considering the above discussion it is straightforward to see that there is a certain overhead and complexity added when autoconfiguration is used in ad hoc networks for ensuring unique IP addresses.

2.2. ID/Locator Based Protocols. A number of proposals for ID/Locator split architecture are in evolutionary stage. Such proposals use identifiers for establishing communication between the entities. ILSA based architectures however require infrastructure components for location indirection. Therefore such proposals cannot be directly used in ad hoc networks. IDHOCNET design is inspired from ILSA in using identifiers for establishing communication between the entities. The basic model of ILSA is shown in Figure 2.

A receiver can insert its location in Internet indirection infrastructure (I^3) and the sender can send the data intended for the receiver without knowing its location. The sender only sends the data and knows the identifier of the receiver. Each entity is identified by its unique identifier whereas location is maintained by an indirection infrastructure which performs indirection based upon the identifiers of the communication entities. The difference among the various competing proposals is in the way identifiers are represented, indirection infrastructure is implemented, and the means by which ID and locator mapping is carried out. Such architectures are

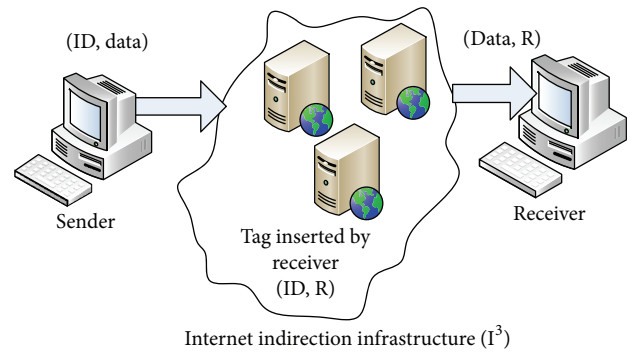


Figure 2: An abstract view of ID/Locator split based architecture.

in evolutionary stages and only small scale testing/prototype implementations are done in this regard.

For elucidation purpose we provide brief details of such proposals and kind of mechanisms used by them for various purposes.

Mobile Oriented Future Internet (MOFI) [23] proposed a distributed mapping control of ID and locators. MILSA (Mobility and Multihoming Identifier Locator Split Architecture) [24] proposes the use of composite form of identifiers. An identifier as per MILSA is composed of flat part which uniquely defines an object within a particular domain whereas the hierarchical part defines the unique domain for which a user is a part. Realm Zone Bridging Servers (RZBS) are used for mapping between identifiers and locators. Enhanced MILSA architecture proposes the use of a composite ID as a combination of human readable and cryptographic ID as a single node identifier. A virtualization architecture based on ID/Locator split concept was proposed in [3]. The 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) [25, 26] approaches the ID/Locator split concept by introducing an extra layer between network and transport layer. This layer is the identity sublayer. The function of this layer is to support the session identification for the upper layers. Security features for the proposed infrastructure are based on asymmetric keys. An identifier overlay network (ION) [27] over IP based networks was proposed for supporting mobility. A clean slate approach was proposed in [28] which is based on 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. Another ID based clean slate approach has also been proposed in [29]. Use of IMSI number as identifiers was proposed in [30] for identity overlay networks.

2.3. IP Address as Identifier. In IP centric architecture the IP address is used as the core identifier, as shown in Figure 3, where IP based MANET is formed using OLSR. The IP addresses are used in the protocol header for establishing sessions, running services, and routing the IP packets in the

```
*** 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 3: Route establishment by OLSR.

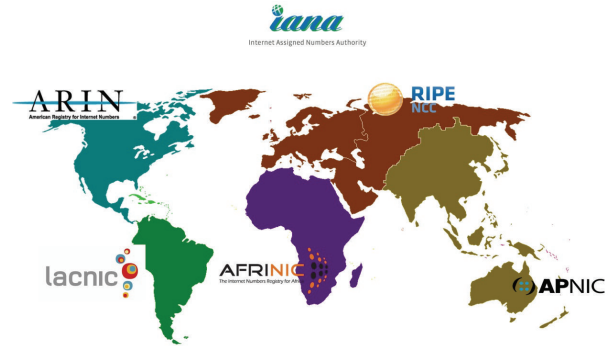


Figure 4: Global view of IP address assignment authorities.

```
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       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 5: Path establishment using MAC addresses.

network. In IP based ad hoc networks, especially in mobile ad hoc network (MANET), nodes cannot have dedicated or same IP address throughout their life cycle. Nodes will be required to change their addresses in an automated manner as dictated by the autoconfiguration service. Such events of address changes will cause the communication disruption between the nodes and this certainly is not a desired network attribute. After the change of IP address network wide broadcast and changes are required for ensuring future communication.

As a practical example consider a scenario shown in Figure 3 in which OLSR is running as a routing protocol.

A user has the view of the current topology and sees that it can communicate with another user, if a priori information about destination IP address is known. But we know that IP address may be changed by the user itself. Moreover, due to the presence of an IP address autoconfiguration scheme, there is likely a chance that IP address of the user will be different from the intended one. Therefore, IP address in such scenario cannot be reliably used for establishing communication.

2.4. IP Address as Locator. Internet is a hierarchal network. The enforcement of this hierarchy is accomplished by Internet Assigned Numbers Authority (IANA) which delegates series of IP addresses to its suborganization known as Regional Internet Registries (RIRs) shown in Figure 4. Each RIR is responsible for its respective region assign IP addresses to various Internet Service Providers. Therefore a host on Internet bears the topological binding in the form of IP address received through its respective ISP. Therefore IP address structure signifies a hierarchical topology and it has been designed specifically for infrastructure based networks. However, in ad hoc networks and especially in MANET the topology is highly dynamic and changes quickly overtime. Moreover the IP addresses are likely to change frequently. Thus due to variation in the assigned IP address to the interface IP address cannot be reliably used for routing packets to the peers. Moreover in case of changes in the IP address of the peer due to address conflicts the routing table entries will also become stale. The change in such cases will be a complex process to handle. Another proposal for mobile hosts to retain their IP addresses while they roam from one network to another is in the form of Mobile IP (MIP) [31]. However MIP also requires infrastructure components of home agent (HA) and foreign agent (FA) for its functionality. Therefore MIP is also not directly applicable in ad hoc networking context.

2.5. Using Identifier for Path Establishment. IEEE 802.11s [32] is an extension or amendment to the IEEE 802.11 standard. It has been designed keeping in view the requirements of ad hoc networks like MANET. As shown in Figure 5, the path is established using MAC addresses of the communication entities. The MAC address in the above case is used for path establishment between the peers and it is responsible for forwarding the data packets. However in the above case the peers still need to use IP addresses for running the applications and in case there is a change of IP address of a particular peer the communication will be disrupted.

2.6. Naming and Name Resolution. IP address is not a suitable choice for identification of nodes. As shown in Figure 6, both IPv4 and IPv6 addresses are difficult to remember. Moreover due to dynamicity of ad hoc networks IP address changes; therefore, the node identification is not persistent. Hence, it is necessary to identify nodes using other types of identifiers. In the present literature of ad hoc networks, it is observed that assignment and resolution of names are carried out by various mechanisms [9]. For realization of naming and name resolution in ad hoc networking scenario nodes are required to exchange control packets which cause overhead for their functioning. Moreover, the role of such services is to map an IP to a name. It is apparent that IP address assigned to the node is variable. Moreover the name assigned to the node is also variable. Therefore, to uniquely identify a node in such paradigm will require double efforts, that is, one to ascertain unique IP address and the other to ascertain unique name.

2.7. Support of IP Based Applications for Backward Compatibility. A completely clean slate design without the backward compatibility of IP based applications will offer less flexibility and lower chances of adoption by the community. Therefore,

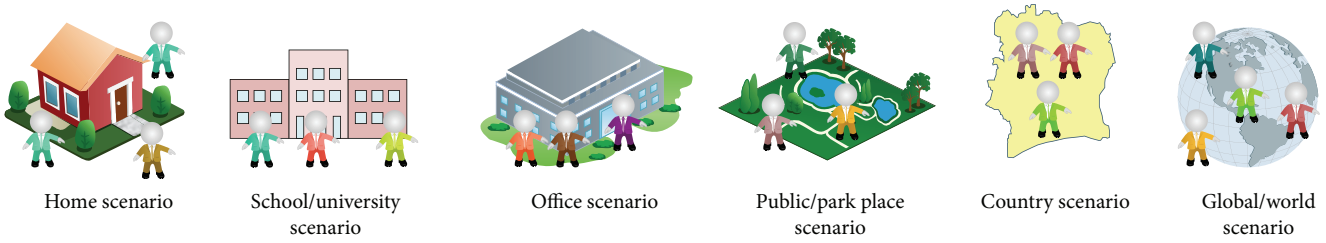


Figure 6: Real world ad hoc networking establishment scenarios.

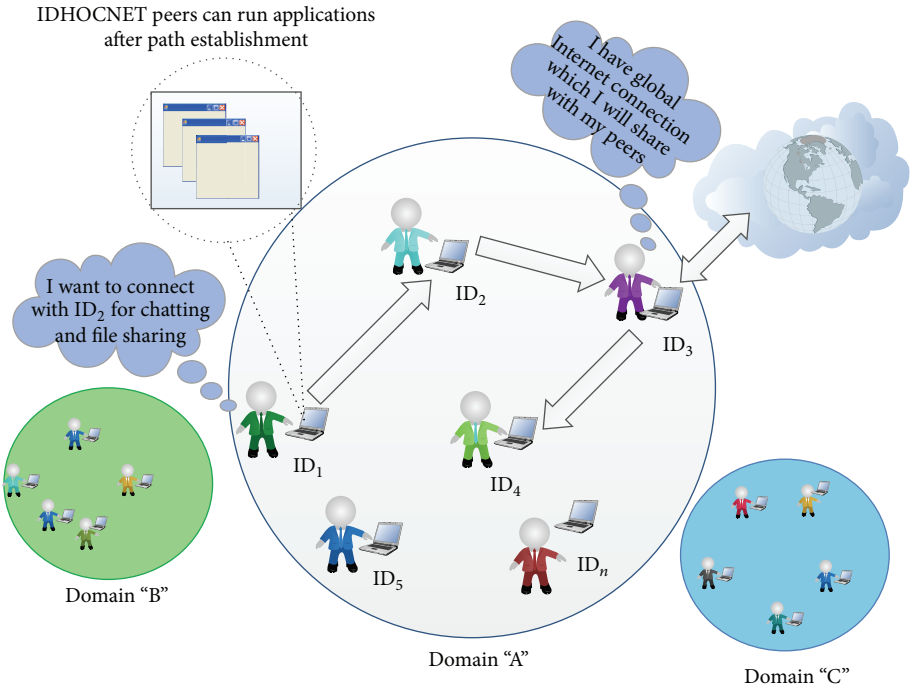


Figure 7: Abstract view of IDHOCNET formation in different domains.

any new architecture should support the standard IP architecture until the maturity of the new architecture. However, provisioning of IP oriented services should be done in such a way that known issues of the IP architecture could be minimized.

3. Architectural Elements of IDHOCNET

3.1. Domain or Realm Perspectives. In the proposed architecture a network is established in a domain. A domain or realm is a group of communication entities which share common context and same identifier class. In a particular domain a host uses a unique identifier to establish communication with its peers. The choice of identifier class rests with the domain in question. The scale of the domain or realm can be small or large and depends upon the total number of identifiers. Various realms are shown in Figure 6. There can be home scenarios where names can be used for identifying respective users. In domains like school, universities, or organizations identifiers like roll number or office identification numbers can be used. In larger domains like public area or countrywide scenarios identifiers like national identification numbers can be adopted. In much larger scenarios identifiers like global telephone numbers, MAC addresses, IMEI, or IMSI numbers can be used.

As discussed above, IDHOCNET function in a particular domain and users can adopt a particular identifier class to communicate with each other. In a particular location there is a possibility of forming IDHOCNET networks pertaining to different domains as shown in Figure 7. There exists a possibility of allowing interdomain communication. However, in this work we will discuss the communication in a single domain. Each domain uses a unique domain identifier. Security policies can be enforced as per requirement of a domain. However, discussion on security provision and implementation is not the scope of the present work.

3.2. Identifiers Based Communication Model. An identifier is a unique entity in the context of particular ad hoc network. In the strict sense an ideal identifier for use in the architecture is a hardware ID or a burned-in address (BIA) which can be translated in packet headers for establishing

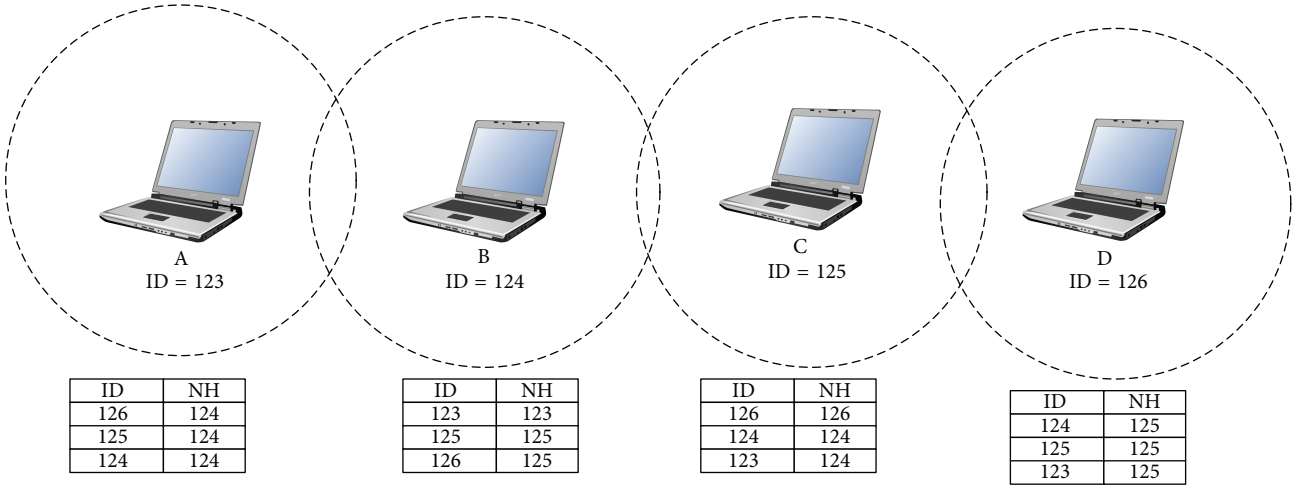
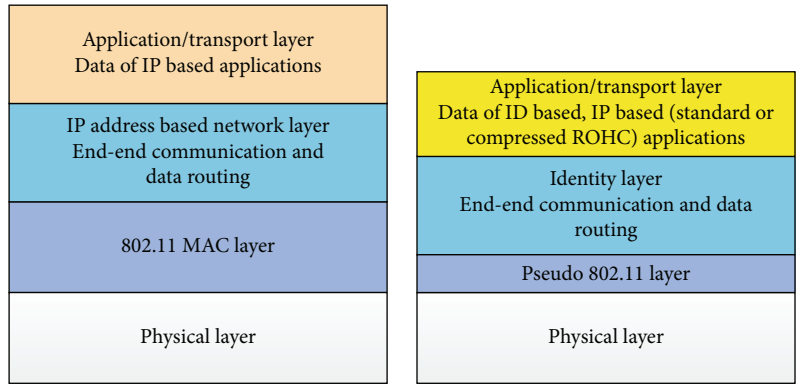


Figure 8: Identifier based path establishment in IDHOCNET.



(a) IP based architecture (b) IDHOCNET

Figure 9: IP Vs IDHOCNET protocol stack comparison.

the communication. As an example, the architecture can use MAC address as an identifier as it is easily translatable to an ID. As an example an arbitrary MAC address, say, 11:22:33:44:55:66, is a hexadecimal number and its decimal equivalent is 18838586676582. It can be observed that the identifier looks similar to a telephone number which can be remembered or written down in the directory for communicating with the desired party. There are other hardware based identifiers like SIM number, IMSI, IMEI, and so forth. Moreover, there is a proliferation 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, and so forth.

The identifiers are used for identification of the entities and used for forwarding the data to other entities. As shown in Figure 8, after path establishment process, an entity can forward the data to the destined entity on the basis of next hop identifier.

3.3. Protocol Stack of the Architecture. ID centric architecture in comparison with IP based architecture is shown in Figure 9. The proposed ID centric architecture uses minimal pseudo 802.11 header which is added after the radiotap header [33]; thereafter identity layer is added. The minimal 802.11 header is added to device driver requirement [34]. The identity layer is used for end-to-end communication and packet forwarding. Moreover, this layer is also used for identification of the communication entities. The last layer is the application or data layer which contains either IP or ID based application data. The communication interface of the system is based on the Wifi device which is commonly available in all smart phones and laptops. The device is required to be in monitor mode for use by the ID based architecture. Each wireless frame is transmitted by inserting radiotap header which is the standard for transmitting and receiving Wifi frames [34].

3.4. Frame Structures of Architecture. The system uses various frames, shown in Figure 10. The basic message M is composed of using radiotap header of 9 bytes and minimal pseudo 802.11 header of 4 bytes. A realm identifier is included for realm identification. After these headers, ID based frame is

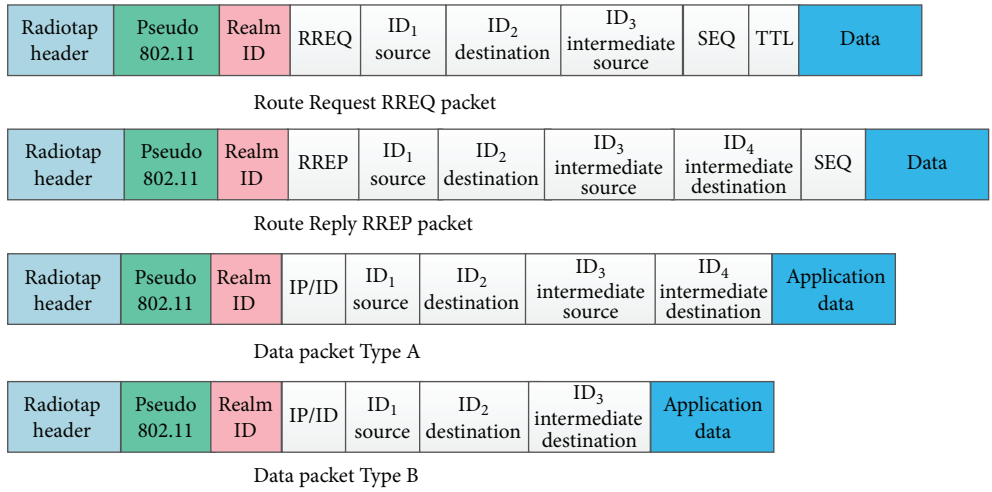


Figure 10: IDHOCNET frames structure.

Table 1: Protocol types of IDHOCNET.

Protocol type	Description
0x00	IP based applications
0x01	Route Request (RREQ)
0x02	Route Reply (RREP)
0x03	ID based applications

constructed. An ID based frame includes 2 bytes for protocol type. Presently available protocol types shown in Table 1 include Route Request (RREQ), Route Reply (RREP), IP applications, and ID based chat application frames. More frame types can be added to the system as per requirement, after the protocol type field identifiers are added to the frame. The size of the identifiers can vary as per the requirement of the domain or realm. A domain or realm is a collection of ad hoc nodes sharing a common context. For a smaller domain or realm which has limited users smaller size identifiers can be used. In order to suffice the requirement of large scale domain like Internet, 16-byte identifiers can be used. Source and destination identifiers are mandatory identifiers and remain same throughout the life cycle of the frame. For all IP/ID based data frames, intermediate source and intermediate destination identifier are added by each node which forwards the frame to the next node. However, it is possible to exclude the intermediate source identifier to conserve the bandwidth. Type A data packets include intermediate source identifier whereas Type B packets do not have intermediate source identifier.

3.5. Processing Steps at Node. Processing steps after reception of a packet at node x are shown in Figure 11. There can be a Route Request (RREQ) or a Route Reply (RREP) packet. Respective actions of RREQ and RREP are elaborated in the path establishment process. If a data packet is received and the next hop destination is known to node x then the packet is relayed to the next hop node. In case the packet is destined

```
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 11: Processing steps of received packet M in IDHOCNET.

for the consumption of node x itself, it is processed as per the application in context.

3.6. Placement of IDHOCNET in Host OS. An IDHOCNET node requires support of operating system to perform its core functionalities. IDHOCNET exploits the varieties of socket interfaces to achieve its operations. Figure 12 shows the placement of IDHOCNET in a host operating system. The system supports ID as well as IP based applications. Elaboration of socket interfaces is provided in Section 5.2.

3.7. Services and Capabilities Announcement. Each host has its own capability and services to offer. The peers may share the information of their capabilities and services through RREQ and RREP messages. Data portion of RREQ and RREP messages contains the capability information. The data of the capability string is defined to let the peers know the types of services being offered. All the intermediate nodes, which process the RREQ and RREP messages, also acquire the services and capabilities information.

3.8. Identifiers Based Application Design. The architecture supports a novel application class known as identifier based applications. This class of applications does not require the use of TCP/IP protocol stack for functioning. An ID based application interacts with the ID server through an

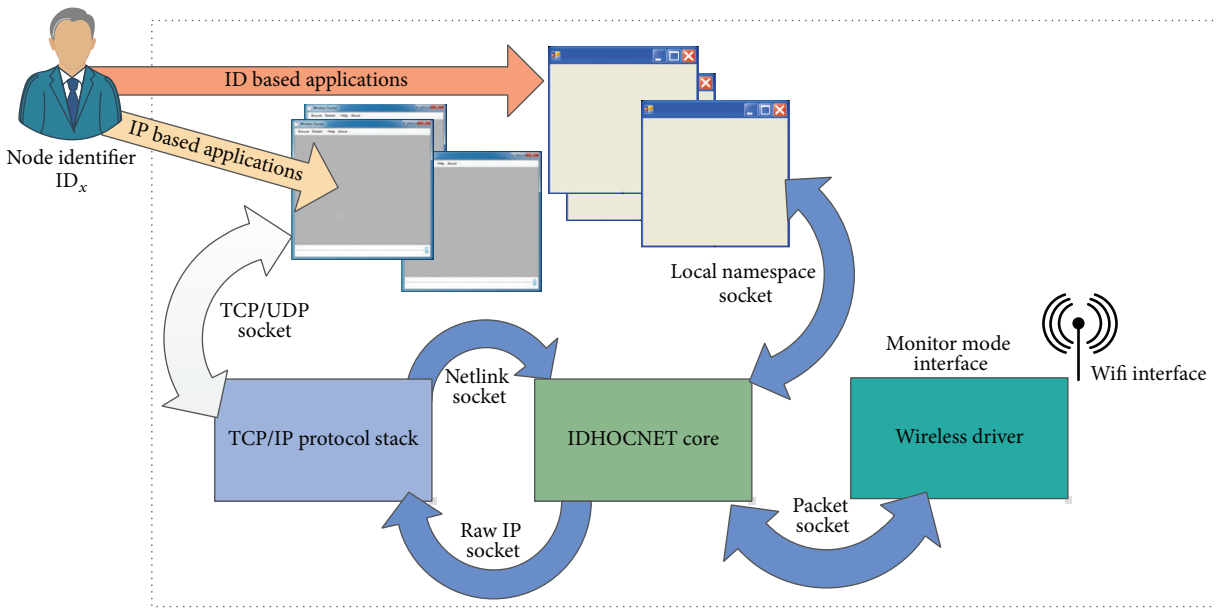


Figure 12: Placement of IDHOCNET in operating system.

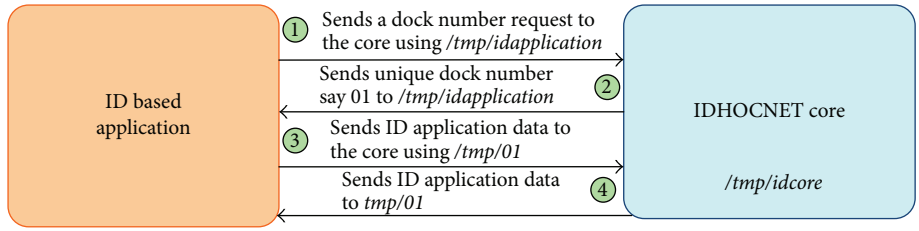


Figure 13: Identifier based application interaction.

Interprocess Communication (IPC) method known as local namespace or Unix domain sockets. Novel ID based applications can be designed which can achieve various tasks like chat, voice, ping, and so forth. The ID server application uses a local namespace socket name, say, */tmp/idcore*. Each category of ID based application holds a unique protocol number as discussed in the frame types section. At startup, the application uses its unique local namespace socket name, say, */tmp/idapplication*, for communication initially with the ID based server application. ID based server application maintains counter of numbers known as dock numbers. These dock numbers are analogous to the port numbers which are used by IP based applications. The flow of ID based application registration with ID based server application is shown in Figure 13.

At step 1, ID based application requests a dock number from ID based server application. ID server application assigns a unique dock number and registers an entry of application protocol number versus dock number in the ID applications table. At step 2, the server ID based application sends a unique dock number to the ID based application instance. After receiving the dock number ID based application deletes its native namespace socket so that other similar applications can initiate additional instances if required. The ID based application uses the dock number as its local namespace socket throughout its life cycle. At step 3, ID based application sends data to the ID server application. At step 4, ID server application sends data to the respective name space socket. Further illustration of the ID based network application flow is available in Section 4.4 under ID Based Application Process Flow.

3.9. Localized IP Networking. In order to support contemporary IP based applications, each node initializes a software defined tap interface. The IP address of the main tap interface (e.g., vnic) acts as the source IP address for all the outgoing IP packets. Whenever a node establishes a contact with a peer, the ID server initializes an alias networking interface and assigns it an IP address. Each pair of main tap interface and the alias network interface acts as a point-to-point network link. Thus a logical Ethernet network as shown in Figure 14 is available at each node for IP application provisioning. Such an implementation of private addresses is also beneficial for ensuring IP address uniqueness and therefore address autoconfiguration service is not required.

As shown in Figure 14, ID based networking architecture has Netfilter hook mechanism which captures the outgoing traffic of each private IP address whenever the user uses an

Table 2: User accessible information of network peers.

ID	Name	Next hop	Private IP	Capabilities
1234	Athar	1235	10.0.0.2	ID chat
1236	Aqeel	1235	10.0.0.3	Internet Gateway, @ port 3120, ID chat
1237	Fahad	1235	10.0.0.4	VoIP, ID chat
1238	N/A	1238		N/A
1235	Shahrukh	1235	10.0.0.5	ID chat

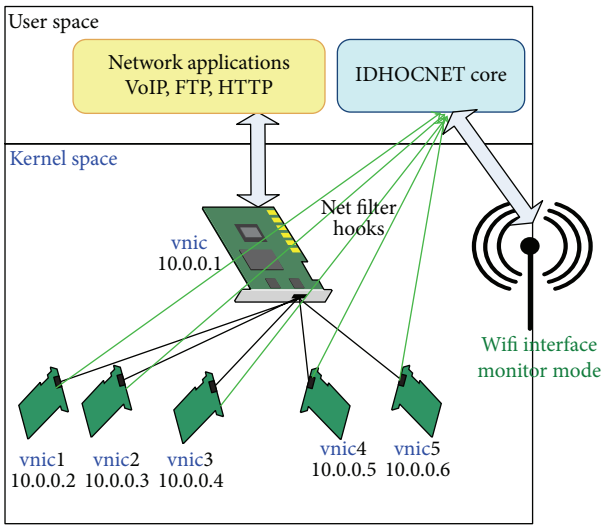


Figure 14: Localized IP address mapping with identifiers.

IP application. For all the outgoing traffic source IP address remains the same which is the IP address of the vnic peer, whereas the destination IP address varies as per the target user. The outgoing IP based data is captured by the ID based architecture due to the presence of Netfilter hooks. Afterwards an ID based header is added to the IP packet and it is further transmitted by the IDHOCNET node.

3.10. ID versus IP Address Binding. A binding with ID and the private IP address is available in the routing table. Furthermore, ID server application also registers an entry in the host configuration file of the operating system. After this entry, IP application can also be run by using the identifier of the remote host. This is possible because IP applications can resolve host IP address through their names available in the host configuration file through gethostbyname() call. Details of the bindings and process flow of IP application are explained in Section 5 under IP based application process flow.

3.11. User View of the Network. Each user of the ID based network has the view of the connected nodes. This view is populated on the basis of routing table and other peer related information. This view is acquired by the user by sending the command to the ID based server application. A user can see all those nodes which are in direct communication and also those nodes for which the user acts as data forwarding node. This view also gives the information about the private IP addresses of the peers which has been assigned locally by the ID based server application running at the user node. A sample view of a user with ID, say, 1230, is shown in Table 2.

4. Operational Flows of the Architecture

4.1. Path Establishment Process Flow. In general, the proposed architecture can adapt any reactive or proactive approach (like OLSR or AODV) for path establishment from TCP/IP model by replacing IP address with real identifier. Initially, for the proof of concept an ID based path establishment procedure similar to AODV is used with a number of simplifications for establishing the path between peers. A path or Route Request is initiated whenever a node requires sharing application data with its peer. RREQ initiating node knows the identifier of node to which connection is required. When node A wants to communicate with node D it initiates a Route Request toward node D as shown in Figure 15. When node A transmits the RREQ it is received by node B at reference point 1; at this point node B initiates a reverse path in its routing table so that path to node A gets registered. Similarly node B forwards the Route Request and it is received by node C at point 2 which registers the reverse path for node A. Now node C transmits the Route Request packet and it finally reaches the destination node D at point 3 and registers the path to reach node A. At this instant node D additionally performs 03 in number functions which include instantiating a private IP address, adding an iptables rule for this private IP address and IP versus ID binding is added to host configuration file.

Now node D transmits the Route Reply (RREP) towards node A. In the Route Reply path the packets are forwarded in the unicast manner because path till node A is known. The RREP transmitted by node D first reaches node C at point 4. At point 4 node C also registers path to reach node D. Node C transmits the RREP which reaches node B at point 5. At point 5, node B registers path to reach node D. Finally the RREP transmitted by node B reaches the final destination node A which performs following functions register of the path to reach node D. Similar to the 03 functions performed at node D, node A also instantiates a private IP address for node D, adds iptables rules, and adds an IP versus ID binding in its hosts configuration file. After the path establishment process nodes A and D can run IP or ID based applications.

4.2. IP Application Process Flow. Let us consider a scenario wherein after the path establishment process between node

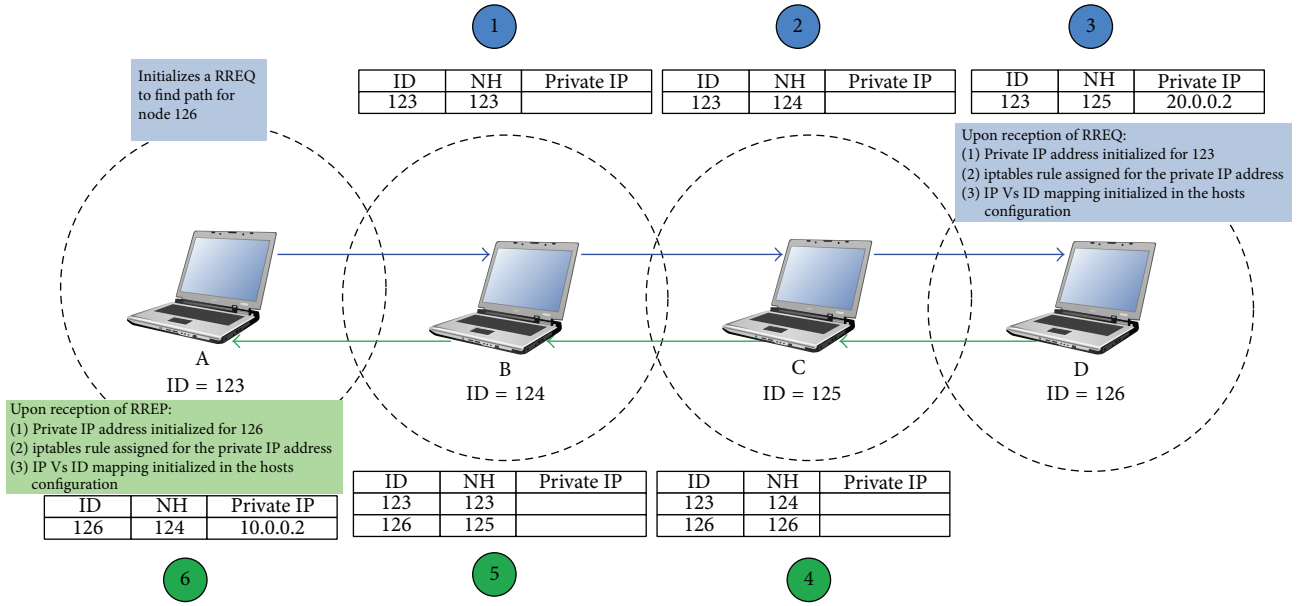


Figure 15: Path establishment process between nodes.

A and node D node A starts an IP based application. The application runs between the IP address of the main tap interface with IP address, say, 10.0.0.1, and the synonym interface IP address configuring node A, say, 10.0.0.2. As depicted in Figure 16, at reference point 1 the IP based application generates the IP based data. The data is captured by the associated TCP/UDP socket. The socket passes the data to the protocol stack at reference point 2. At reference point 3, the IP packet is captured by the packet interception system of the ID based server application. At reference point 4, the routing table is referred and identifiers are added. At reference point 5, the packet moves to the packet socket system after composition of wireless headers. At reference point 6, the network driver writes the packet to the Wifi air interface. At reference point 7 the packet leaves the air interface where the intermediate nodes forward the data to the destination node. When the packet reaches the destination node at reference point 8 it is received by the air interface. At reference point 9, the packet socket captures the data and wireless header is stripped off. At reference point 10, whether the packet is destined for the node is checked. At reference point 11 IP addresses compatible to the node are associated with the packet; that is, destination address is changed to main tap interface address, that is, 20.0.0.1, and source address is changed to 20.0.0.2. At reference point 12 raw socket injects the packet back into the protocol stack. At point 13, the TCP/UDP socket captures the packet. At point 14 the active IP application consumes the packet.

4.3. Internet Accessibility Process Flow. Users of the ID based network may also have access to the Internet. A gateway node as shown in Figure 17, with an additional wired or wireless interface is used for Internet provision. This node runs a light weight proxy server and listens to the main tap interface IP address at a specific port address, say, 3120. This proxy server forwards the HTTP requests for all the ID based peers to the Internet bearing interface. These requests are captured by the monitor interface and injected to the protocol stack by the ID server application. After the injection of HTTP request packet, an HTTP response is received. The HTTP response is captured by the ID server application due to the configured net filter rule for outgoing IP packets. Each response is encapsulated in ID based packet and transmitted back to the requesting nodes. The users configure their browser's proxy setting with their locally assigned IP or with the identifier of the proxy node. They know the availability of the proxy nodes through the RREQ or RREP processing. When the packet is received at the requesting node then the ID based server application injects the packet and the user can see the response in its Internet browser.

4.4. ID Based Application Process Flow. After the path establishment process users can run ID based applications like ID based chat, ID based ping, and so forth. Figure 18 shows the process flow of the ID based application run. Initially both end points start their ID based chat application. The application at both ends registers in the respective ID based server application with application type versus dock number entry. Initially, node A does not know the destination dock number of chat application running at node D. At step 1, node A sends the message to node D with the source dock number 03 and destination dock number as –1. When node D receives the message the ID based application sees in the ID based application table the type of frame and the respective dock number. The server application forwards the data to application available at dock 1. Now node D knows the node A dock number of chat application. At step 2 node D sends message using source dock number 01 and destination dock

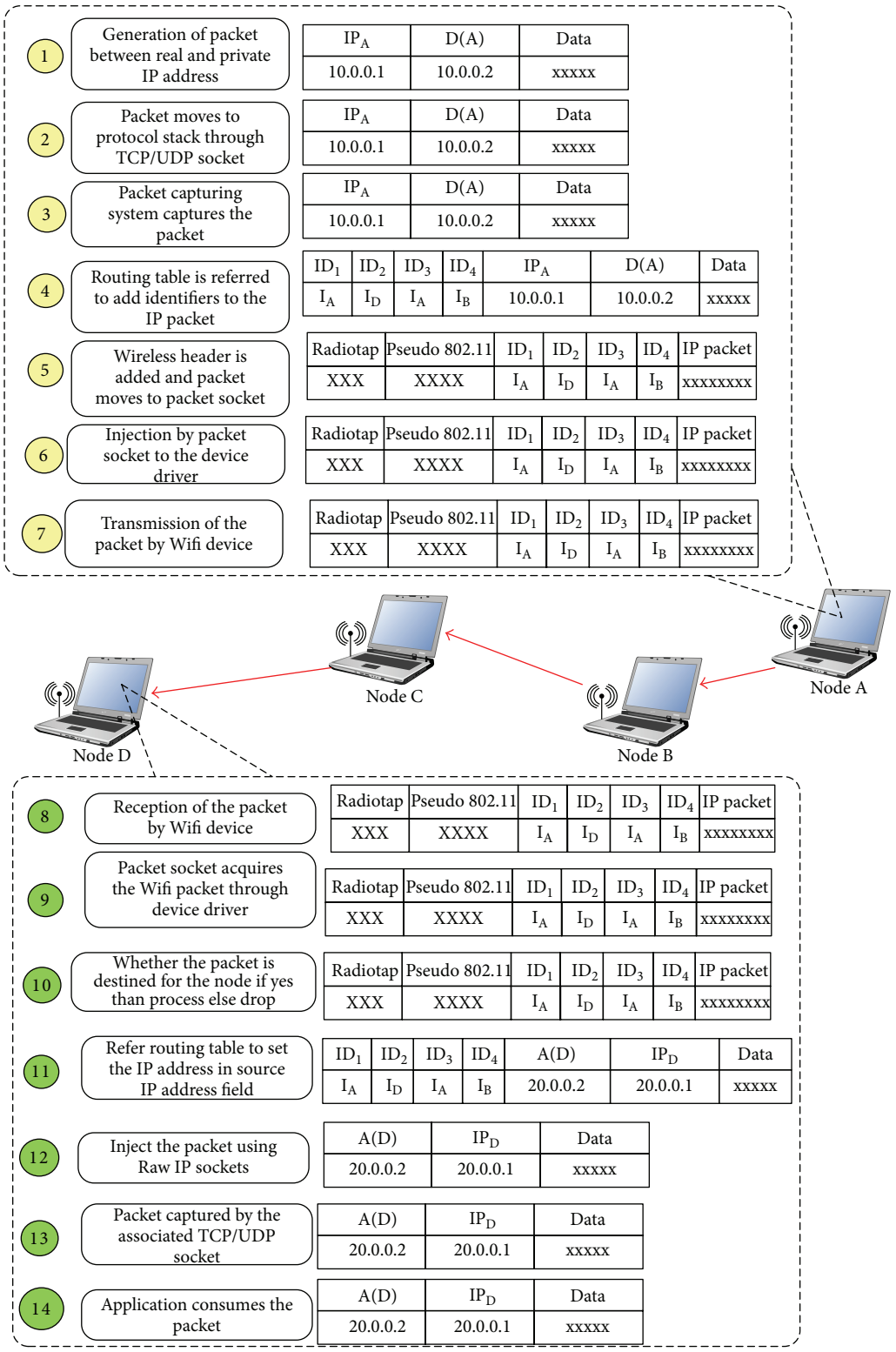


Figure 16: IP application packetwise flow.

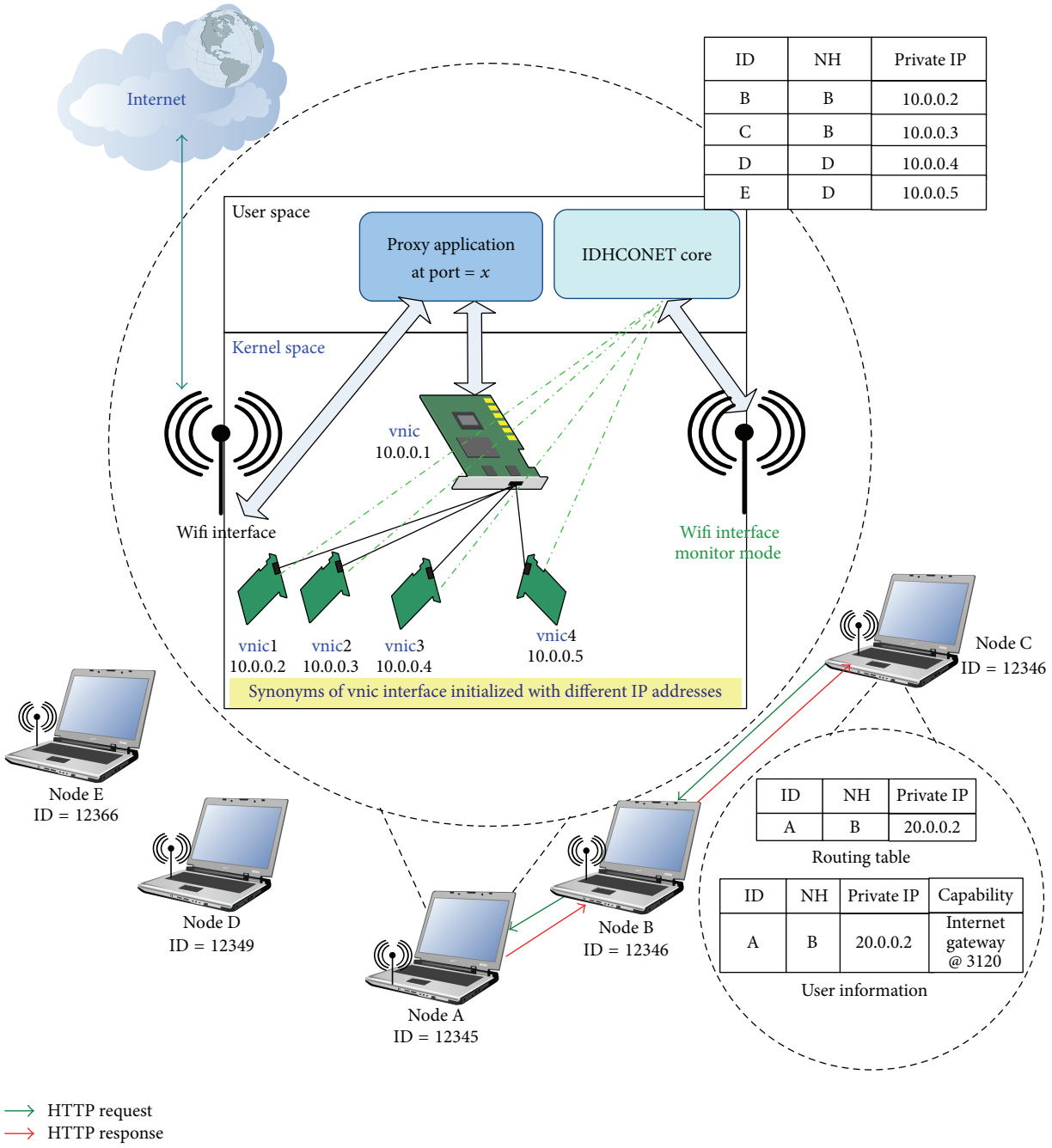


Figure 17: Internet provision through gateway node.

number 3. When A receives the message it now also knows the dock number of node D chat application. Now both node A and node D can exchange messages.

5. Implementation Details

5.1. Host Configurations. Laptops with Ubuntu 12.04 LTS operating system were used to build a host. C++ has been used as the development language. In order to ensure that each host has same software configuration, RemasterSys tool [15] was used. The image supports features like 80211.s mesh networking and OLSR. TL-WN721N device of TPLink was used as Wifi USB interface due to its 802.11s driver support on Linux. However, the proposed system has no restriction and it was tested to work on a number of Wifi devices. The size of the identifiers is 8 bytes which is sufficient to accommodate the IMEI, MAC addresses, global telephone or mobile numbers, and so forth.

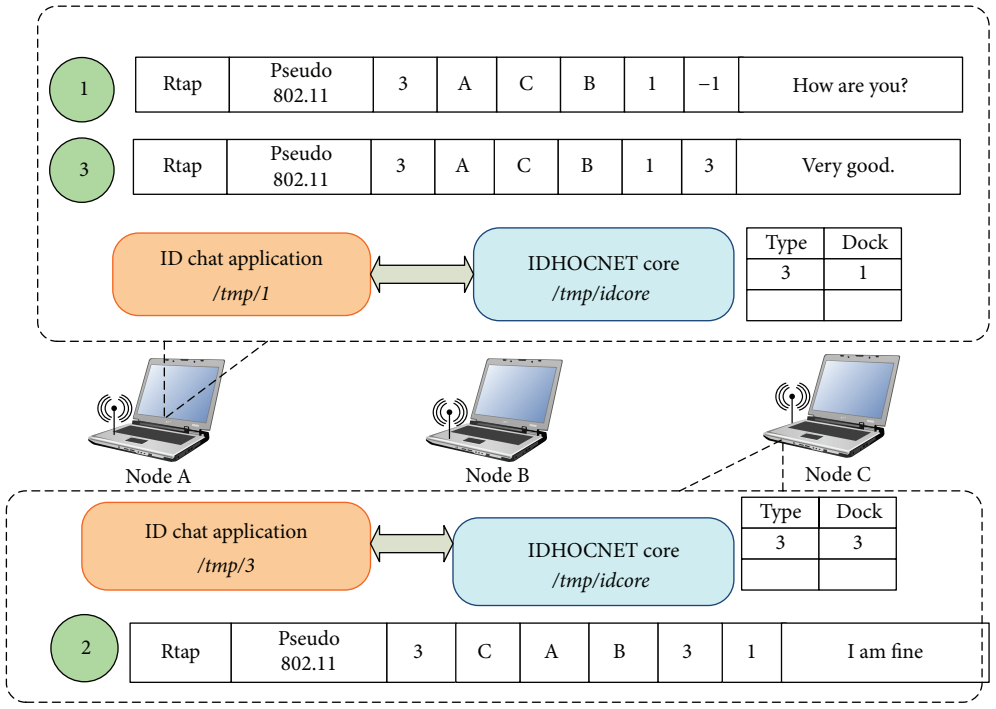


Figure 18: ID application process flow.

Table 3: User accessible information of network peers.

Socket type	Functionalities
Packet socket	Capturing and transmitting Wifi Packets
Raw socket	Injection of IP packets to local IP addresses
Netfilter socket	Capturing of IP packets against iptables rule
Namespace or Unix domain socket	Intercommunication of ID based application

5.2. Sockets Application Programming Interfaces. The implementation uses different types of sockets API for performing various functionalities and communication provision. Table 3 gives the type and functionality of the socket APIs.

5.3. Packet Capturing System. For supporting the IP based application a Linux Netfilter system [16] is implemented using LibnetFilterQueue and Libnf NetLink [17]. An ID based application automates the process of adding an iptables rule against every private IP address added for the peer by issuing a Linux system command. When a particular packet is received in the callback function of the Netfilter system NF_DROP verdict is issued to further stop the flow of the packet. Identifiers based header is added to the captured packet so that it can be forwarded to the destination node. There are two possibilities that we can either send the uncompressed packet to the destination nodes or an optional ROHC based IP header compression can be applied to the received packet flow. The option is set in the configuration parameters of the software whether ROHC compression is required or not.

5.4. Interfaces and Private IP Addresses. The main tap interface (e.g., vnic) and monitor mode interface (e.g., monIf) are initialized at the start of the system through Unix system commands. When a node has active communication with other nodes then virtual interfaces are added to it (e.g., vnic : 1, vnic : $2, \ldots,$ vnic : n). A snap shot of interfaces of IDHOCNET is shown in Figure 19.

6. Experimentation, Experiences, and Results

6.1. Testbed Configuration. Multihop configuration between end points was established for test IP and ID based applications. All IP and ID based applications were tested up to 5 hop configuration settings. All the hosts were configured with identical hardware and software configuration as mentioned in Section 5.1. Channel 10 of Wifi was selected for all testing and applications run. The data was collected by using Wireshark tool.

6.2. P2P IP Based Applications Execution. Various IP based applications were tested. After path establishment between the end points, a user issued a ping command for private IP address 10.0.0.2. Moreover, the user was also able to identify the destination directly. In case the identifier is used

```
shahrukh@shahrukh:~$ ifconfig
monIf     Link encap:UNSPEC  HWaddr 10-FE-ED-19-EA-C1-00-00-00-00-00-00-00-00-00-00
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1580 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:510374 (510.3 KB)  TX bytes:0 (0.0 B)

vnic      Link encap:Ethernet  HWaddr b2:07:5d:9c:69:3c
          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:500
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

vnic:1    Link encap:Ethernet  HWaddr b2:07:5d:9c:69:3c
          inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1

vnic:2    Link encap:Ethernet  HWaddr b2:07:5d:9c:69:3c
          inet addr:10.0.0.3  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1

vnic:3    Link encap:Ethernet  HWaddr b2:07:5d:9c:69:3c
          inet addr:10.0.0.4  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
```

Figure 19: Private address map view.

Figure 20: Receiving server response through a web browser.

its IP is resolved by gethostbyname() call available in the IP based applications. A regular VoIP application [18] was tested successfully. The application [18] was based on ALSA library and OPUS codec. A multihop scenario of 04 hops was established for verifying the voice reception between the end points. After the ID based path establishment process between the peers, the voice between the end points was transmitted and received successfully and the two parties were able to clearly hear each other.

6.3. Client Server Based IP Application Execution. The framework supports Client Server paradigm of IP based architecture. Apache web server is configured at a node and an application is hosted which tells the current temperature of the surrounding by reading the sensor value. The application was accessed successfully from a client browser by typing the identifier of the server as shown in Figure 20.

Moreover, a gateway node with Internet connectivity was added and Internet was accessed in multihop configurations.

6.4. P2P ID Based Application Execution. A simple ID based application is designed using the principle and techniques outlined in the architecture details. In Figure 21(a) user with ID 18687085636098 is connected with ID 18687085636099. The multipath is established using the ID based path establishment procedure outlined above. The user sends a message to the destination. After traversing the intermediate nodes the message is received successfully at the destination node as shown in Figure 21(b).

6.5. Data Packet Forwarding Overhead Comparison. Transmission size of a frame encapsulating an IPv4 packet of 20 bytes with 30-byte payload under different forwarding mechanism is shown in Table 4. The size of 802.11s link layer based forwarding is highest due to the use of IP addresses and additional MAC addresses. The size of IP layer based forwarding is the same as that of Type A of the IDHOCNET. In order to efficiently conserve the bandwidth Type B header can save an additional 08 bytes. As forwarding is based on identifiers and IP addresses are set at reception and as per the scheme at the node there is a possibility for not sending the source and destination IP addresses in Type A and Type B headers thus saving an additional 08 bytes in case of IPv4 addresses.

Table 4: Data packet comparison using different protocols.

Forwarding type	MAC	IP	Payload	Total
Link layer	64	20	30	114
IP layer	52	20	30	102
Type A	52	20	30	102
Type B	44	20	30	94
Type A (No src, des IP)	52	12	30	94
Type B (No src, des IP)	44	12	30	86

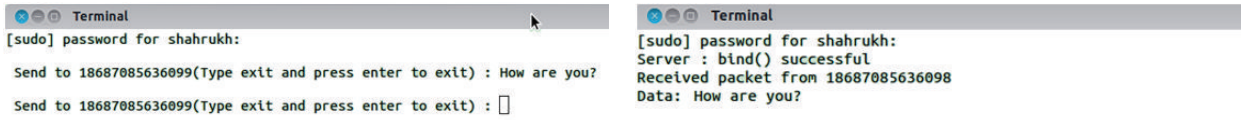
6.6. RTP/UDP/IP Stream Analysis. In order to estimate the VoIP statistical performance we established a multihop testbed setup, complying with the host configuration discussed in Section 5.1. In each configuration RTPSEND [19] application was used with constant bit rate traffic at 20 msec and constant payload size of 168 bytes. For each configuration, 5000 RTP/UDP/IP packets were collected using OLSR, 80211.s, and finally our ID based protocol with Wireshark tool. ROHC library was used for compression of the RTP/UDP/IP packet when ID based protocol was used. The size of the complete frame also including the radiotap header is shown in Figure 22.

The packet loss ratio was measured during each scenario which is shown in Figure 23. It was observed that 802.11s suffers when more hops were added. The performance of IP layer forwarding and our proposed system using Type A or Type B headers is almost similar.

The jitter performance is shown in Figure 24. The jitter values calculated in different hops reveal that 802.11s performance is worst among the different protocols. The performance of Type A and Type B is better than the IP layer forwarding.

In order to appreciate the amount of bytes saving by the proposed ID based system in comparison to Layer 3 and Layer 2 forwarding by IP based architecture transmitted bytes were collected over the period of three minutes. Figure 25 shows the total transmitted bytes during the total span of 01, 02, and 03 minutes.

After the lapse of three-minute interval the scheme successfully saves considerable amount of bytes. In comparison with Layer 2 forwarding, Type A saves 7560 bytes, whereas Type B saves 9000 bytes. In comparison with Layer 3 forwarding Type A saves 5400 bytes whereas Type B saves 6840 bytes. It is evident that the ID based protocol is an ideal candidate for usage in real time streaming services like VoIP.



(a) Sending data to the user

(b) Receiving data at the remote end

Figure 21: P2P ID based application sending and receiving data.

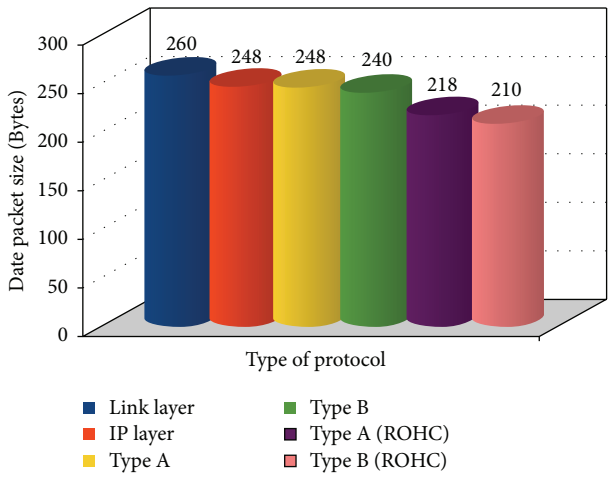


Figure 22: Frame size of RTP/UDP/IP packet under different protocols.

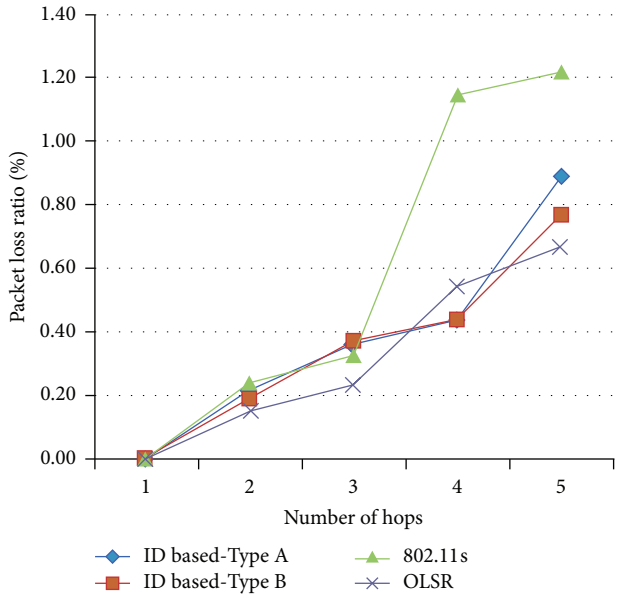


Figure 23: Packet loss ratio under different scenarios.

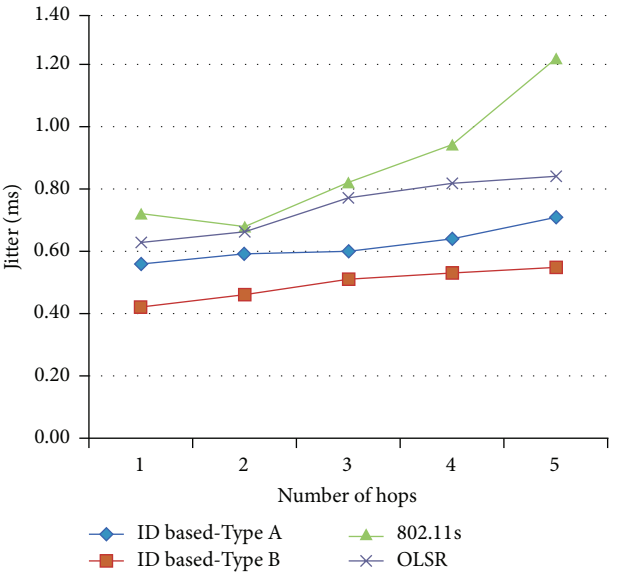
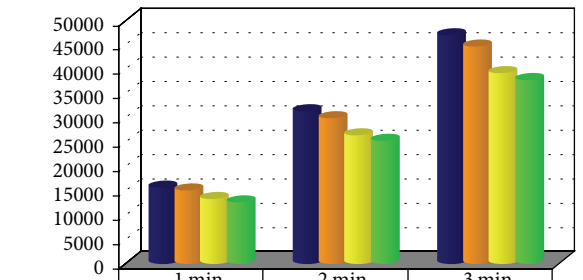


Figure 24: Jitter under different scenarios.



	1 min	2 min	3 min
Link layer	15600	31200	46800
IP layer	14880	29760	44640
Type A	13080	26160	39240
Type B	12600	25200	37800

Figure 25: Total transmitted bytes under different scenarios.

7. Comparison of Framework with Existing Ad Hoc Network Realization

The proposed framework provides number of benefits over the existing realization of ad hoc networks based on IP based networking architecture. A comparison is presented in Table 5. In IP based architecture the nodes use IP addresses as end point identification. The usage of the IP addresses is temporary in nature and required to be changed due to address conflicts. Moreover in terms of user point of view, the structure of the IP address is difficult to remember, whereas the proposed ID based framework can use real world identifiers like telephone number, mobile number,

Table 5: Comparison between IDHOCNET and IP based ad hoc networks.

Requirement	Existing ad hoc schemes	Proposed scheme	Merit over IP based networks
End point identification	Uses IP addresses as identifiers.	Supports identifiers like global telephone numbers and MAC addresses.	Identifiers like telephone numbers, mobile numbers can be used.
Name resolution	For naming support it requires the naming and name resolution services [9] for identification support.	Uses identifiers (IDs) for end point identification, no need for resolution requirement throughout network.	No overhead for name resolution thus saving network bandwidth.
IP address autoconfiguration	These schemes require an IP address autoconfiguration service [10–12] to run in the ad hoc network.	It uses private address map for resolving IP address autoconfiguration. The proposed scheme does not suffer from IP address conflicts.	No requirement to run an IP address autoconfiguration scheme thus saving network bandwidth.
Heterogeneous subnets	Special gateway nodes are required for communication between nodes belonging to different subnets.	IP address support is implemented using private address map managed by the node. Therefore each node can have its own subnet.	No requirements for special gateway nodes.
Uniformity	Different schemes for IP address autoconfiguration assume different roles by different nodes like root node, group leader [21], and proxy node [22].	All nodes have similar role in the architecture.	Provides design simplification due to similar role.
Multihop header compression	Simulation studies [35, 36] have been conducted for multihop ROHC in ad hoc contexts. However a practical multihop ROHC implementation does not exist.	Implements a multihop Robust Header Compression scheme.	Supports a practical multihop Robust Header Compression (ROHC) scheme which saves considerable amount of network bandwidth.

MAC address converted into its numeric equivalent, and so forth. As the users are already familiar with such types of identifiers, therefore transition for using identifiers instead of IP addresses will be straightforward and advantageous in terms of ease of use.

The existing ad hoc networking requires additional distributed name assignment and name resolution services [9] for naming support. Addition of such schemes adds complexity and communication overhead. The proposed framework does not require naming and naming resolution thus saves the additional overhead required by the existing ad hoc networking schemes. IP address autoconfiguration service is an essential requirement for existing ad hoc networking scheme for ensuring unique IP addresses throughout the network. The service adds complexity and communication overhead, whereas the proposed scheme implements private addresses at each node; therefore IP address autoconfiguration scheme is not required. Nonusage of IP address autoconfiguration will save bandwidth.

In existing ad hoc networking all nodes come under similar subnetwork for communicating with each other. In case of provision of communication between different subnets special gateway nodes are required. This constraint will become dominant in case of mobile ad hoc network if two networks are initialized with different subnetworks. In case of mergers all nodes must come under same subnetwork or the provision of special gateway nodes must be available. In comparison to IP based networks, the proposed scheme can have disjoint subnets due to the private IP address usage at each node.

A number of naming, name resolution, and IP address autoconfiguration schemes demand special roles from ad hoc network nodes. In case of failure of special purpose node the functionality is difficult to maintain, whereas in the proposed ID based scheme all nodes have similar role.

To the best of our knowledge there is no practically implemented multihop IP header compression scheme for IP based ad hoc networks. However, simulation studies for MANET IP header compression (MIPHC) and MIPHC/ROHC schemes [20] have been conducted. These schemes offer less bandwidth conservation as compared to point-to-point ROHC scheme, whereas the proposed framework implements a multihop IP header compression using ROHC library, with similar compression gain offered by ROHC for point-to-point links.

8. Conclusion and Future Work

An ID based architecture has been designed and realized for ad hoc networks which is likely to open future research directions. Problems of IP address autoconfiguration along with dual role of IP address as an identifier and locator can be solved by using our proposed methodology. We have provided support for operation of contemporary IP based applications. A novel class of network application known as ID based application has been designed and tested in

multihop scenario. The VoIP scenario has been tested using multihop testbed using OLSR, 80211.s, and ID based protocol. The overall overhead of ID based protocol is found to be lower than IP based protocols. Moreover, the protocol supports ROHC in multihop scenario which will be highly desirable in ad hoc contexts. An overall comparison of our architecture with existing ad hoc networking shows visible benefits. In future, we aim to extend our work by implementing a kernel module for the proposed architecture. The present work does not encompass the security features which is an essential requirement for ad hoc networks; our future work will focus on security provisioning of our ID centric architecture.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] J. Postel, "Internet protocol," RFC 791, 1981, https://tools.ietf.org/html/rfc791.

[2] R. Jain, "Internet 3.0: ten problems with current internet architecture and solutions for the next generation," in *Proceedings of the IEEE Military Communications Conference (MILCOM '06)*, pp. 1–9, IEEE, Washington, DC, USA, October 2006.

[3] C. So-In, R. Jain, S. Paul, and J. Pan, "Virtualization architecture using the ID/Locator split concept for Future Wireless Networks (FWNs)," *Computer Networks*, vol. 55, no. 2, pp. 415–430, 2011.

[4] 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.

[5] 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," *Computer Networks*, vol. 60, pp. 13–33, 2014.

[6] R. Droms, "Dynamic host configuration protocol," Internet RFCs RFC 1541, 1997.

[7] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, "Dynamic host configuration protocol for IPv6 (DHCPv6)," RFC 3315, 2003.

[8] P. Mockapetris and K. J. Dunlap, "Development of the domain name system," *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 4, pp. 123–133, 1988.

[9] M. Masdari, M. Maleknasab, and M. Bidaki, "A survey and taxonomy of name systems in mobile ad hoc networks," *Journal of Network and Computer Applications*, vol. 35, no. 5, pp. 1493–1507, 2012.

[10] L. J. García Villalba, J. García Matesanz, A. L. Sandoval Orozco, and J. D. Márquez Díaz, "Auto-configuration protocols in mobile ad hoc networks," *Sensors*, vol. 11, no. 4, pp. 3652–3666, 2011.

[11] S. Khalid and A. Mahboob, "A survey on Auto-configuration mechanisms for Mobile Adhoc Networks (MANETS)," *Journal of Engineering and Applied Sciences*, vol. 31, no. 2, 2013.

[12] H. Zhou and M. W. Mutka, "Review of autoconfiguration for MANETs," in *Wireless Ad-Hoc Networks*, H. Zhou, Ed., chapter 6, pp. 123–144, InTech, Rijeka, Croatia, 2012.

[13] 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.

[14] U. Ghosh and R. Datta, "A secure dynamic IP configuration scheme for mobile ad hoc networks," *Ad Hoc Networks*, vol. 9, no. 7, pp. 1327–1342, 2011.

[15] S. Rafiul and H. Subrata, "SAAMAN: scalable address autoconfiguration in mobile ad hoc networks," *Journal of Network and Systems Management*, vol. 19, no. 3, pp. 394–426, 2011.

[16] H. Kumar and R. K. Singla, "Architecture for address autoconfiguration in MANET based on extended prime number address allocation (EPNA)," *WSEAS Transactions on Computers*, vol. 8, no. 3, pp. 549–558, 2009.

[17] L. J. G. Villalba, J. G. Matesanz, A. L. S. Orozco, and J. D. M. Díaz, "Distributed dynamic host configuration protocol (D2HCP)," *Sensors*, vol. 11, no. 4, pp. 4438–4461, 2011.

[18] L. Javier, G. Villalba, J. García, A. Lucila, and S. Orozco, "An extension proposal of D2HCP for network merging," *Journal of Ubiquitous Systems and Pervasive Networks*, vol. 3, no. 1, pp. 35–40, 2011.

[19] W. Xiaonan and Z. Shan, "An IPv6 address configuration scheme for wireless sensor networks based on location information," *Telecommunication Systems*, vol. 52, no. 1, pp. 151–160, 2013.

[20] A. Zimmermann, A. Hannemann, and B. Schleinzer, "IP address assignment in wireless mesh networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 3, pp. 321–337, 2011.

[21] M. F. Al-Mistarihi, M. Al-Shurman, and A. Qudaimat, "Tree based dynamic address autoconfiguration in mobile ad hoc networks," *Computer Networks*, vol. 55, no. 8, pp. 1894–1908, 2011.

[22] X. Wang and H. Qian, "A tree-based address configuration for a MANET," *Pervasive and Mobile Computing*, vol. 12, pp. 123–137, 2014.

[23] J.-I. Kim, H. Jung, and S.-J. Koh, "Mobile oriented future internet (MOFI): architectural design and implementations," *ETRI Journal*, vol. 35, no. 4, pp. 666–676, 2013.

[24] 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," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 1–6, IEEE, New Orleans, La, USA, November-December 2008.

[25] 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.

[26] V. P. Kafle and M. Inoue, "Locator ID separation for mobility management in the new generation network," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 1, no. 2-3, pp. 3–15, 2010.

[27] Y. Wang, J. Bi, and X. Jiang, "Mobility support in the internet using identifiers," in *Proceedings of the 7th International Conference on Future Internet Technologies (CFI '12)*, pp. 37–42, Seoul, Republic of Korea, September 2012.

[28] P. Martinez-Julia and A. F. Gómez-Skarmeta, "A novel identity-based network architecture for next generation internet," *Journal of Universal Computer Science*, vol. 18, no. 12, pp. 1643–1661, 2012.

[29] B.-O. Kwak, T.-H. Lee, and W. Chun, "ID based communication in domain-insulated autonomous network architecture

(DIANA)," in *Proceedings of the International Conference on ICT Convergence (ICTC '12)*, pp. 264–269, IEEE, Jeju Island, South Korean, October 2012.

[30] Y. Wang, "UNA: a new internet architecture for user-level multihoming and mobility," in *Proceedings of the 6th International Conference on Future Internet Technologies (CFI '11)*, pp. 13–18, Seoul, Republic of Korea, 2011.

[31] C. E. Perkins, "Mobile IP," *IEEE Communications Magazine*, vol. 35, no. 5, pp. 84–99, 1997.

[32] R. C. Carrano, L. C. S. Magalhães, D. C. M. Saade, and C. V. N. Albuquerque, "IEEE 802.11s multihop MAC: a tutorial," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 1, pp. 52–67, 2011.

[33] G. H. Berg and L. R. Johannes, "Radiotap," http://www.radiotap.org/.

[34] M. Vipin and S. Srikanth, "Analysis of open source drivers for IEEE 802.11 WLANs," in *Proceedings of the International Conference on Wireless Communication and Sensor Computing (ICWCSC '10)*, pp. 1–5, Chennai, India, January 2010.

[35] B.-N. Cheng, J. Wheeler, B. Hung, S. Moore, and P. Sukumar, "A comparison of IP header compression schemes in MANETs," in *Proceedings of the IEEE 32nd International Performance Computing and Communications Conference (IPCCC '13)*, pp. 1–9, San Diego, Calif, USA, December 2013.

[36] B.-N. Cheng, J. Zuena, J. Wheeler, S. Moore, and B. Hung, "MANET IP header compression," in *Proceedings of the IEEE Military Communications Conference (MILCOM '13)*, pp. 494–503, IEEE, San Diego, Calif, USA, November 2013.