Computer Networks 55 (2011) 415–430

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Virtualization architecture using the ID/Locator split concept for Future Wireless Networks (FWNs)

Chakchai So-In [b], Raj Jain [a,*], Subharthi Paul [a], Jianli Pan [a]

[a] Department of Computer Science & Engineering, Washington University in St. Louis, One Brookings Drive, Box 1045, St. Louis, MO 63130, USA
[b] Department of Computer Science, Faculty of Science, Khon Kaen University, 123 Mitaparb Rd., Naimaung, Maung, Khon Kaen 40002, Thailand

ARTICLE INFO

Article history:
Available online 22 September 2010

Keywords:
ID/Locator split
Future Wireless Networks
FWNs
Next generation wireless networks
NGWNs
Mobility
Multihoming
Privacy
System Architecture Evolution
SAE
Virtual object
Object
Virtual object to virtual object communication
Virtual channel
Channel
Multi-tier
Future wireless internet
Future internet
Network architecture

ABSTRACT

Future Wireless Networks (FWNs) will be a convergence of many fixed and mobile networking technologies including cellular, wireless LANs, and traditional wired networks. This united ubiquitous network will consist of billions of networkable devices with different networking interfaces. A common networking protocol is required to communicate among these devices and interfaces; System Architecture Evolution (SAE) documents state that Internet Protocol (IP), world-widely used in the current Internet, is likely to become that common protocol. However, traditional IP architecture has faced several known challenges, such as mobility, multihoming, privacy, path preference selection, etc., which should be resolved in FWNs. One of the difficulties in the current IP architecture is the overloading of IP addresses used both as the identity and the location of IP devices. In this paper, we propose a virtualization concept for networkable components, or *(virtual) objects*, which generalizes all abstract components to potentially be used in FWNs. In addition, we have explicitly separated the functions of the virtual object identity from the virtual object location (using the ID/locator split concept). The end-to-end communication is a *concatenation* of the involved components, called a *channel*. To help support the ownership and policy enforcement for trusted vs. untrusted networks, a set of (virtual) networkable components with the same interest, called a *realm*, is formed in a multi-tier structure. The individual policy can be enforced for each individual group of (virtual) objects and/or channels. This virtualization architecture concept, characterized by the ID/locator split concept, is well-suited for FWNs and helps eliminate problems in the current Internet.

1. Introduction

Future Wireless Networks (FWNs) offer a large-scale *interoperability* of diverse traditional wireless networks with many types of wireless technologies: cellular networks, sensor networks, RFID (Radio-Frequency IDentification) networks, and the conventional wired networks. FWNs are evolving into an ubiquitous network in which customers or users will not need to be aware of the different behaviours and/or characteristics of the networking media underneath their applications [1]. Moreover, a policy-based control would be necessary to make use of multiple interfaces [2–5] in an efficient way.

FWNs should also support peer-to-peer, point-to-multipoint, and ad hoc infrastructure modes. FWNs may provide a *guaranteed service* with an agreement on the quality of service (QoS) control, as well as best-effort services. In addition, the emergence of billions of networkable mobile wireless devices, which may outnumber the wired PC's as early as 2010 [4], including Laptops, PDAs (Personal Digital Assistants), cell phones, wireless sensors, etc., shall exacerbate the problem of *scalability* in the current networks.

* Corresponding author.
E-mail addresses: chakso@kku.ac.th (C. So-In), jain@cse.wustl.edu (R. Jain), pauls@cse.wustl.edu (S. Paul), jp10@cse.wustl.edu (J. Pan).

1389-1286/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2010.08.013

Moreover, with the advances in networking technologies, the concept of a single user-single host-single interface will no longer be common in FWNs. Users bearing several multi-interface wireless devices leveraging a variety of networking interfaces, such as wireless local area networks (WLANs), 2G/3G, LTE (Long Term Evolution), (Mobile) WiMAX, and Ethernet shall call for an ubiquitous high-speed networking environment that can inherently support *mobility*: mobility over large geographic topologies and mobility of users (mobile users) over devices, device *multihoming* and concurrent multi-interface sessions.

With various networking connections, service providers and mobile users should be able to choose the best connection (*path preference selection*) based on cost and quality of service (QoS) requirements. Multiple interfaces should allow load sharing, load balancing, and higher availability with recommended path information from the service providers. Also, the mobile users should be able to maintain their *privacy*, while the networking environment should provide inherent *security*. Apart from all these requirements, FWNs' designers also need to evaluate the transition steps from the current networks to FWNs, e.g., how to incrementally *deploy* the FWN system into the current network [6–9].

The issues of interoperability, guaranteed service, scalability, mobility, multihoming, path preference selection, privacy, security, deployability, etc., discussed above, represent some of the key requirements for the design of FWNs. Given these different sets of requirements, it is quite difficult to predict which direction FWNs will be headed, especially in terms of a common communication protocol among networking components. The 3rd Generation Partnership Project (3GPP) has made a decision to adopt Internet Protocol, or IP [10,11], into cellular networks as well. System Architecture Evolution (SAE) is the core networking architecture being developed by 3GPP [12–14] for the next generation of cellular wireless networks. SAE will be an *all-IP based mobile wireless network*.

Note that FWNs will face the same problems that have been identified for the current Internet. The Internet now is not only being used academically, but also in industry with a non-trustworthy design for commercial applications. So, this design has brought difficulties into the relationship amongst the organizations and the administrative hierarchies. More importantly, one of the greatest issues of the current IP architecture is *the overloading of IP address semantics* [15–19]. The IP address acts as a host or node identifier as well as a locator in the routing space. This contextual overloading implicitly binds a host to its point-of-attachment in the network, and there is no independent namespace to represent the end host itself. Thus, every time the end host moves to a new network or changes its interface; and consequently obtains a new IP address, all the sessions bound to the previous IP address are broken.

Such an implicit overloading makes it difficult to support full mobility, multihoming, traffic engineering, privacy, security, etc. As a result, in this paper, we propose a new concept on how to apply the ID/locator split idea into the IP-based FWNs. In addition, we extend this splitting concept beyond hosts in order to be general enough to cover all feasible physical and logical components, or *objects*, in FWNs. We call this the *virtualization of objects*.

Note that in this paper, we do not intend to limit the architecture to a specific solution, but rather provide the virtualization architecture concept in general. Obviously, there are possible solutions available; some may meet the requirements, and some may not. Nevertheless, we include some probable techniques when we introduce the architecture requirements.

This paper is organized as follows. In Section 2, we discuss common terminologies in the traditional network architecture. In this section, we compare an illustration of a wired/wireless and cellular network structure. Also, we briefly explain a cross-over function among these terminologies. In Section 3, we discuss the proposal of an ID/locator split concept that will apply to the virtual networkable components in FWNs. Using examples, we illustrate how to apply the FWN architecture concepts to our current network in Section 4. In Section 5, we show feasibility by applying the ID/locator split concept into our virtualization architecture. In Section 6, we briefly describe related work focusing on the ID/locator split concept proposed in the current IP networks. We also briefly point out their pros and cons which leads to our proposal. Finally, our conclusions are drawn in Section 7.

2. Current networks: terminology and system architecture

This section describes terminologies used in current networks. We also discuss a conceptual definition for each term and notation with provided examples. In addition, we discuss two main current network architectures illustrated by examples: wired/wireless data networks (Internet) and cellular networks.

2.1. Terminology

Name: a word or a combination of words, readable and recognizable by humans, to identify a person, place, or thing, such as *John Smith*,*Washington University in St. Louis*, *Intel*, and *Microsoft*. Usually, *name* is also represented by the organizational management, which tends to be hierarchical; for example, *john_smith.cec.eng.wustl.edu* represents user John Smith in the Department of Computer Science and Engineering, School of Engineering and Applied Science, Washington University in St. Louis.

Address: a point of attachment or the name of the place where a person, something, or organization may normally be reached; for example, *One Brookings Drive*, St. Louis, MO 63130 USA is the address of Washington University in St. Louis.

Locator: where something could be located currently, such as GPS (Global Positioning System) latitude and longitude positions. Note that the address and the locator are very similar, and in some contexts they are the same. For instance, *One Brookings Drive*, St. Louis, MO 63130 and GPS positions at 38° 38′ 52.82″N and 90° 18′16.22″W are considered as both the address and the locator. However,

in a Mobile IP environment [20,21], a home IP address can be represented as the address, and its Care of Address (CoA) is the locator.

Identifier (ID): a representation of a particular person or a thing. The identifier is usually unique within a particular domain, called *Local ID*. That local ID may result in a global unique ID with a combination of local ID and domain ID, called *Global ID*. For example, a student ID, 388812, is an unique ID within the Washington University in St. Louis domain. A combination of University ID and student ID is globally unique. Similarly, a telephone number, 233-7456, is unique within the city of St. Louis and the state of Missouri. With the prefix 314, the identifier is unique within the USA and globally unique with the additional prefix of 1.

Name vs. Identifier: Name and identifier represent the same object; however, a name is basically readable by humans and easier to remember and recognize. Usually, there is a one-to-one mapping relationship between name and ID. For example, the user name *John Smith* has a social security number or his identification number 498-21-3611. There are also optional alias names and/or nicknames. Note that this relationship implicitly represents his ownership and existence in the world. For this particular example, this ID is unique within the USA, or we can say John Smith is within the US domain.

Another example is an explicit hierarchical naming system which is related to his ownership and function, such as *John.Smith@wustl.edu* and *John.Smith@intel.com*. With these examples, *John Smith* is tied to some domains and can usually be provided with his unique ID within an organization. In the first example, the identifier (Social Security Number) is permanent for his life; however, in the second example, the mapping can be changed over the organizational policy.

Hardware vs. Software: IDs can be used to represent both hardware and software. Hardware IDs represent the identity of the physical device, such as MAC (Media Access Control), used as the identifier of Ethernet networking interfaces; IMEI (International Mobile Equipment Identity), used as the identifier of GSM phones (Global System for Mobile communications); and IMSI (International Mobile Subscriber Identity), used as the identifier of SIM cards (Subscriber Identity Module).

For software IDs, considering a TCP/IP (Transmission Control Protocol/Internet Protocol) stack, the IP address may represent the identifier at the network layer, and the combination with TCP port number is the identifier at the transport layer. Note that in some contexts, these IDs are used interchangeably. For example, the MAC address can also represent the identifier for the link layer in the TCP/IP stack (also used as the networking interface ID). So, this MAC address resents both hardware and software IDs.

Tier Structure: A logical concept of an arrangement of grouped components within the same specific interests; for example, an user-tier means a representation of a group of users, and a host-tier means a group of hosts. Note that we call a *physical-tier* for a group of hardware as in the previous two examples, and a *logical-tier* for a group of software, such as network and transport layers.

2.2. Wired/wireless network architecture: examples

In traditional wired/wireless networks, no differentiation among *Name*, *ID*, *Address*, and *Locator* exists. Moreover, in some contexts, the functions are overlapped. There are four terms involved in these networks (one physical and three logical): FQDN (node or host name), optional TCP/IP port number, IP address, and MAC address, respectively. The DNS resolution process results from FQDN from/to IP addresses.

The Address Resolution Protocol (ARP) resolution process is used to map MAC addresses from/to IP addresses. In general, the MAC address is globally unique and may be assigned to each individual networking interface. Again, normally there is a one-to-one relationship among these; however, it is also possible for many-to-many relationships for load-balancing/sharing and host-aliasing purposes. In the current wired/wireless networks, these relationships are not well specified.

One example of these networking components in traditional wired/wireless networks is as follows. Considering the host aspect, a host name, *hive.cec.wustl.edu*, is an unique hierarchical FQDN. Its IP address, 128.252.20.98, can be represented as a global address of this host. Within the local domain or LANs (Local Area Networks), the MAC address also represents the host with an assumption that one host consists of only a single interface, e.g., 00-1F-3C-6A-0D-69. In this example, IP and MAC addresses are also used as host IDs. Since the host IP address is used to route the packets, the IP address is also used as the host locator. It is obvious that the use of an identifier and a locator here are redundant and/or ambiguous.

Consider user *john_smith.hive.cec.wustl.edu* as a representation of an unique hierarchical user name corresponding to this particular host; his ID and locator will be tied to the host itself or the IP address. Whenever the host or the user moves, IP addresses change.

2.3. Cellular network architecture: examples

In cellular networks, four terms are commonly used: *User Name*, *Mobile Phone Number*, *IMEI*, and *IMSI*. The first term is used as Name and the others as IDs. Again, usually there is an one-to-one mapping amongst these terms, but not necessarily. For example, the *yellow pages* consist of a simple database to map an user name to his cellular phone number, such as *John Smith* and his phone number, 1 314 555 9191.

The hierarchical structure of a cellular phone number can also represent the real geographical location of the user and the node (host or device). Each device (cellular phone) contains IMEI, a global device identification number, e.g., 49015420323751. In addition, IMSI is another unique identification, but this ID is represented as an individual SIM card.

Table 1 summarizes the traditional wired/wireless/cellular networks and their common terminologies. Considering the tier structure, only two physical tiers are involved: user and host/device. Four logical tiers, DNS Name ID, optional TCP Port Number (relating to user specific applications), IP address, and MAC address, represent

Table 1
Networks (Wired/Wireless/Cellular) vs. Terminology: User and Host/Device.

Networks	Terminology
Wired/Wireless Networks	User (User Name) ↔ Host (Host Name, IP address, and MAC address)
Cellular Networks	User (User Name) ↔ Host/Device (Telephone Number ID, SIM ID, and Device ID)

application, transport, Internet, and link layer abstractions for wired/wireless networks.

Note that we can consider the network infrastructure as one more tier [22]. The ambiguity and dependency of all related terms and functions make it difficult to achieve full mobility, multihoming, location privacy, etc., required for FWNs. In FWNs, these traditional networks will merge into one united ubiquitous network. Therefore, an unique representation and/or function are required. We will discuss all concepts required for FWNs and provide examples in the next section. We will again revisit some of these traditional network terminologies and functions, especially as applied to FWNs in Section 4.

3. FWNs: terminology and system architecture

This section discusses the concepts used for the FWN architecture. We revisit some of required terminologies and relate them to their functions. We introduce a virtualization concept, *virtual object*, of the component, *object*, in FWNs. Then, we represent the communication between the (virtual) objects/components represented as a *(virtual) channel*.

Since each virtual object is independent from others, mobility, multihoming, and location privacy can be directly applied to each individual virtual object. We provide a detailed description of these terms in the next section. These terms are required to ensure FWNs support all possible representations in the future. In addition, the policy enforcement can apply in a particular virtual object, a group of virtual objects, and the layout of virtual objects with quality of service (QoS) controls.

3.1. Terminology introduced in FWNs

Object: an addressable component that can be physical and/or logical. Examples of physical objects are: Personal Computer (PC), Router/Switch, Cellular Phone, Networking Interface, and Human. Examples of logical objects are: Application and Transport Layers. Each object has an identifier (ID) and optional Locator (s) and Name (s).

Virtual Object: a virtual representation of objects; for example, virtual machine (a logical machine that executes like a real machine), virtual interface (a logical networking interface), and virtual network (a logical network that provides a specific set of guaranteed resources shared from a physical network), etc.

(Virtual) Tier: Logical concepts of an arrangement within the same specific interest (virtual) object group.

Multi-Tier: A hierarchically vertical communication tiers.

(Virtual) Channel: (virtual) object to (virtual) object communication; a concatenation of objects or virtual objects. The channel allows the communication establishment of both inter-tier and intra-tier communication.

Virtual Identifier: Especially for privacy purposes, the virtual identifier [4] is a representation of an identifier resulting from multiple levels of ID mapping or other mappings from the identifier to its locator, primarily to hide the actual ID.

Realm: A hierarchical group of virtual components that logically belongs within the same organization. The organization defines its own policy and provides a trusted relationship.

In FWNs, we apply all required terminologies, such as an identifier, locator, address, tier, etc., from the current network. However, *Address* has the same function as *Locator*. The address is only used for location, not identification. Therefore, *Address* is no longer a notation in FWNs. *Name* functions as *ID*, except being readable and/or recognizable by humans. *ID* is usually independent from the locators.

All components are represented as *Objects*. Each object has an identifier and optionally has names and locators. In addition, similar to hardware/software definitions (See Section 3.1), the object represents both physical and logical definitions. The virtual representation of object (s) is called *Virtual Object*. We consider (virtual) object to (virtual) object communication as a *(Virtual) Channel*. Notice that host-to-host and user-to-user communications are just examples. A group of the same interested objects is called *Tier* and *Virtual Tier* for a group of virtual objects.

3.2. Virtual object abstraction

In this section, we discuss in detail the virtual object (VO) abstraction and (virtual) channel or virtual object to virtual object (VO-to-VO) communication used as an end-to-end communication.

3.2.1. (Virtual) object description

A virtual object (VO) represents both logical and physical illustrations of an object. The *virtual object* has an identifier, optional locators, and optional names. Fig. 1 shows an example of a (virtual) object. In this example,

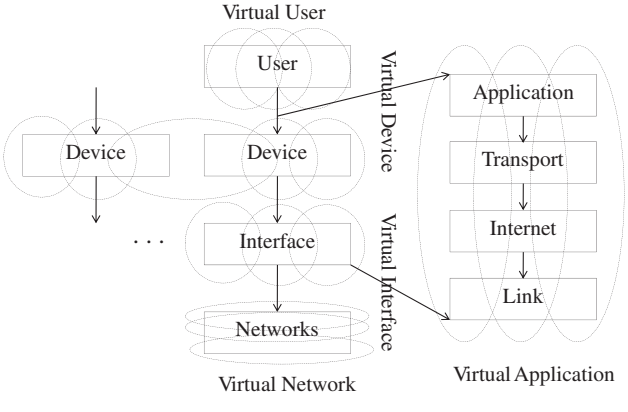

Fig. 1. (Virtual) Object Examples: virtual user, virtual device, virtual interface, virtual network, and virtual application.

user, device, interface, and network are physical objects. Application, transport, Internet, and link are logical objects. Each particular virtual object may consist of several objects, such as virtual devices or virtual hosts, virtual interfaces, virtual networks, and virtual applications.

The virtual objects may share the same resources. For example, many virtual host objects share the same physical device or host object; however, there is a tight boundary of the guaranteed resources allocated to each virtual object for QoS control purposes. The mechanisms to achieve this tight boundary are out of the scope of this paper. The dashed circle and oval shown in Fig. 1 represent each individual virtual object. Note that a virtual object may not be limited to a single physical object. It may span multiple objects; for example, a virtual application may operate over multiple devices (parallel/distribute computing), and a virtual storage host may consist of many networking storage hosts.

3.2.2. (Virtual) Object to (Virtual) object communication or (Virtual) channel

In the virtual object model, the end-to-end communication is established by the layout of a *concatenation* of many (virtual) objects, called *(virtual) channel*. The layout of the communication pattern is not limited to inter-tier communication but also intra-tier communication (See Fig. 2). This (virtual) channel is again treated as an individual communication of a shared resource in which the organizational policy can be enforced with QoS controls.

Fig. 2 shows an example of (virtual) channel representation. Fig. 3 shows the communication point of attachment for each inter-tier and intra-tier communication, called Object Access Point (OAP). We define the input and output of a virtual object as Input Object Unit (IOU) and Output Object Unit (OOU). Consider a cross layer communication. These points of assessment, OAP, IOU, and OOU, allow communication amongst the virtual objects, such as to send some useful information from bottom tier to upper tier. Link reliability information (e.g., for congestion and collision indication separation; and wireless channel characteristics for modulation and coding optimization purposes) can be sent to change the transmission property or characteristic and/or choose a proper transmission

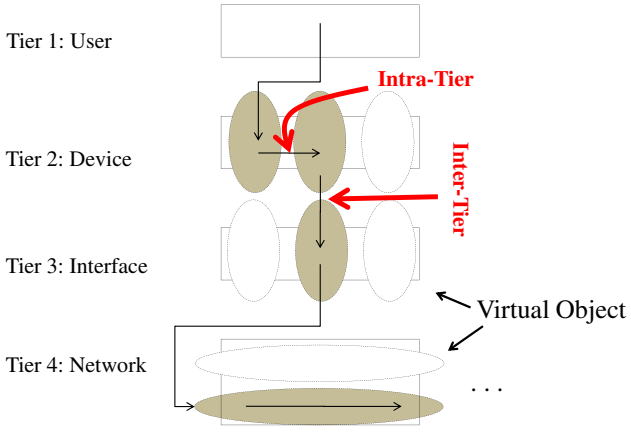

Fig. 2. (Virtual) Channel Examples: Four tiers (User, Device, Interface, and Network) with Intra-tier and Inter-tier communication.

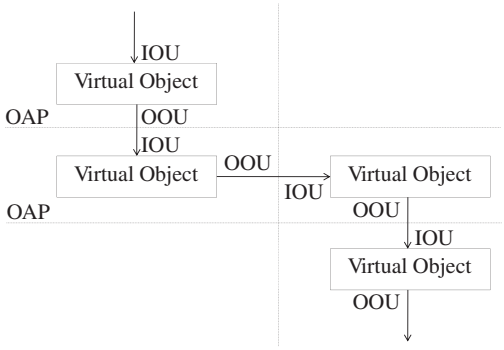

Fig. 3. Object Access Point (OAP): Input Object Unit (IOU) *versus* Output Object Unit (OOU).

channel. Note that the IOU and OOU header overheads are added for each tier communication.

3.2.3. (Virtual) object mobility and multihoming

In the virtual object model, each virtual object has a built-in mobile characteristic (mobile object = object). Since each virtual object is independent and treated individually, mobility and multihoming can directly apply to each individual virtual object. Consider the mobility aspect. Each virtual object has a locally unique identifier (within a particular domain) and locator (s). In some contexts, we can consider the identifier as IOU and the locator as OOU for each virtual object. The locator, or OOU, is usually independent from IOU so that when the virtual object moves, its ID remains the same, but its locator may change.

Applications in FWNs will be established with IDs, not locators. Consider the multihoming aspect. Again, the virtual object abstraction allows one-to-one, one-to-many, many-to-one, and many-to-many relationships of the communication to form the (virtual) channel with individual policy enforcement and QoS controls.

3.2.4. (Virtual) object privacy

In FWNs, *Privacy* can be applied to each individual virtual object: the location and ID privacy are just examples. In fact, the virtual concept to represent the real object is implicitly used in terms of the object privacy. Consider the user location privacy. Since the users no longer require the locators to reach the destination, this can facilitate the user location privacy. Consider an individual ID. Similar to the virtual ID concept in [4], in FWNs, a virtual ID is also used to represent one or more levels of privacy in order to hide the real ID; and again for location privacy, the identifier can be treated as a virtual location.

3.2.5. (Virtual) object path/ (Virtual) channel selection

In a mobile wireless network environment with several devices or interfaces attached to different service providers, the virtual user object should be able to select his own path based on the cost of service and QoS. To support this path or channel selection preference, an agreement among the virtual objects along the path is required, normally with the cooperation of a cloud of service providers to support load sharing/balancing systems so that the systems can forward the transaction to the end point of the virtual object with the preferred path.

The virtual object may use different techniques and information acquired from the path information or from the service provider to select the channel. For example, in [4] a weighting mechanism is used to identify the ingress path to the user. In [5], a policy-based mechanism achieved by applying a linear programming concept is used to select underlying multiple interfaces for multihoming purposes.

3.2.6. Policy enforcement

As described above, the policy management can be directly applied to individual (virtual) objects and channels. The policy is enforced through the concept of realm or domain. A set of policy and QoS control parameters is required for guaranteed services and the virtualization of shared resources. We will describe this issue in detail in the next section.

3.3. Realm managers and mapping system servers

In this section, we discuss the policy management enforced by realm managers. We also describe the functions of the realm servers as well as providing some possible mechanisms existing in the current networks.

3.3.1. Hierarchical policy enforcement

A realm manager mainly functions as a policy enforcement manager for a group of (virtual) objects. The realm managers may be hierarchically distributed for delay latency reduction purposes with an increase in the number of realm servers. For example, in a mobile wireless networking environment, mobile user objects frequently change their locations. This also requires cooperation among realm managers to develop the communication channel and provide the guaranteed resources.

In general, the border router gateway and/or the base station can function as a realm manager. Fig. 4 shows an example of a realm hierarchy. In this example, there are three tiers:L1,L2, andL3. Each realm manager Rx_1x_2 (x_1 being an index of hierarchical tier structure and x_2 representing the realm managers within the same tier) is responsible for policy management in a particular realm. Note that the policy of the lower-tier realm managers cannot rewrite those of the higher-level realm managers.

3.3.2. Location services and decentralized management

A realm manager also functions as a mapping server to resolve the identifier to/from the locators. Again, the realm manager is managed in a distributed and hierarchical manner. A cloud of realm managers within the same realm can also communicate with each other, which helps to mitigate the scalability, load balancing/sharing, and fault-tolerance of the system architecture.

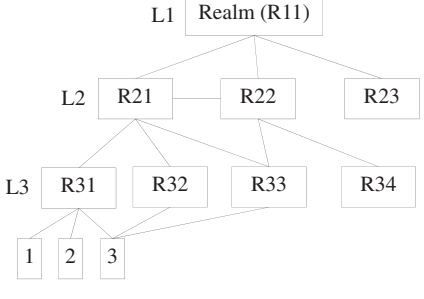

Fig. 4. Realm Hierarchy: three levels.

A mobile object tracking system is required in FWNs, and the realm manager should support this function. The realm manager also cooperates with others to provide the location service. For example, mobile user objects may query as to where the nearest mall and coffee shop are. Note that this location service follows the mobile object's location privacy policy. Several location discovery mechanisms can be used in different contexts/networks, such as location discovery in sensor networks [23] and mobile ad hoc networks [24,25].

3.3.3. Service discovery

A realm manager also provides a discovery service for mobile object users or other objects; for example, the mobile object users can query on and list offered services. Several proposals on service discovery when applied in different networks may be used, such as a multipath cross-layer service discovery in mobile ad hoc networks [26], a community-based service discovery [27], and a context aware semantic service discovery [28].

3.3.4. Realm managers with proxy/relay function

A realm manager can also function as proxy and relay servers. Especially in a mobile wireless environment, mobile objects tend to move frequently at high speeds. In addition, to save battery power, mobile objects are primarily in a sleep mode and wake up only when necessary. Because of these characteristics, the realm manager should keep track of mobile objects and should buffer, and then relay/forward the transactions from/to the mobile objects when they are awake [29]. The proxy can also operate on behalf of legacy nodes unaware of the virtual object concept; this is similar to the use of a proxy in Mobile IPv6 [30].

3.3.5. Realm managers with guaranteed resources and QoS controls

Each virtual object and/or virtual channel established is considered a shared resource. Therefore, the realm manager should provide enough resources, according to the promised QoS, to the virtual object. Consider the virtual channel setup. This requires communication among the realm managers to provide the resources reserved for the entire (virtual) channel. Again, the mechanisms to achieve this tight boundary are out of the scope of this paper; further information can be found in [31–34].

Moreover, especially in a mobile wireless environment, when mobile objects move, the communication among realm managers is also required to guarantee the promised QoS, say from one base station to another. The reservation for the resources for each particular service is based on the agreed policy setup.

To communicate among realm managers, a (virtual) channel reservation control protocol may be used along the path. This reservation control protocol may be similar to Resource ReSerVation Protocol, or RSVP, [35]; however,

this modified RSVP should be aware of the virtual object mobility, multihoming, channel information, etc.

3.3.5.1. Leasing control management. FWNs should support the resource leasing as well. Since each virtual object/channel can be governed by an individual set of policies (each virtual channel with a shared guaranteed resource or a virtual application object may be owned by different service providers), it is possible to manage and to develop a pricing model between service providers and customers. Some of resource leasing concepts can be applied into the architecture, such as a mechanism for resource leasing management for a smart space [36] and for suspending a virtual machine [37].

Another example is that of a virtual network when a small service provider can lease a virtual channel consisting of a set of virtual hosts and virtual network objects in order to provide access to its own application and/or Internet. Again, the small service provider can provide services or sub-lease its set of shared resources. Virtual application leasing is also applicable in FWNs. The virtual application requires the cooperation among realm managers to provide the guaranteed resources.

3.3.6. Multiple ID resolution and mapping database

A realm manager is required to support multiple ID resolution functions. The realm manager stores a mapping database of both inter-realm and intra-realm mapping information. In addition, for privacy purposes, a virtual ID may be used, and the realm manager is supposed to also store the mapping of this virtual ID and the actual ID.

All mapping databases should be stored at realm managers. Within a single domain/realm, several realm managers may work as primary and secondary servers for fault-tolerance. Furthermore, the database can also be stored in Distributed Hash Tables (DHTs) [38] for scalability purposes. Note that each DHT is managed within a particular realm, and the database between different realms is managed in a hierarchical manner.

3.3.7. Realm managers assisted multihoming with multiple objects

A realm manager can also provide the multihoming knowledge database for a virtual object. The realm manager acts as a knowledge-based system to recommend the path (s) to the virtual object. The virtual object can use this information as well as its own policies to make a final path selection.

3.3.8. Other functions of realm managers

A realm represents an administrative domain, or organization. Each organization has several functions that help in the efficient operation of the organization. These functions can be performed by the realm managers. Several such functions are listed below. All of these functions are optional and can be performed by other objects in the system.

3.3.8.1. Dual-stack realm managers. For backward compatibility to legacy nodes, a realm manager also functions in a dual-stack mode so that legacy nodes that do not recognize the virtual object concept and/or an ID/locator separation can communicate with each other in FWNs. The realm manager works as a convertor/encap-decapsulator to forward the transactions from/to FWNs. This concept is similar to the use of Dual stack hosts and router (DSMIPv6) in Mobile IPv6 [39].

3.3.8.2. Virtual realm managers. The realm manager is also an object. So, the virtual realm manager is a virtualized realm manager, e.g., many different virtual realm managers can operate over a single realm manager server, or one virtual realm manager can be spread over multiple realm manager servers.

3.3.8.3. Garbage collection realm manager. The realm manager may keep track of the mobile objects and their usages of resources. It may be possible that the mobile objects acquire the resources, and then leave them unused. The realm manager may periodically check/update the use of resources of the mobile objects [40,41].

3.3.8.4. Channel state controller. To help support the migration of (virtual) objects and/or (virtual) channels, the realm manager may function as a channel state controller and store the channel state information required to re-establish the connection. Additionally, similar to the relaying and buffering functions described earlier, the realm manager may function as a caching server. Whenever there is a connectivity disruption for mobile users, the realm manager may cache the information until the operation is back on track. To achieve the migration, similar approaches to a process migration [42–44], service migration [45], and virtual machine migration [46] can be applied.

3.3.8.5. Auto-reconfiguration manager. A realm manager may help in the establishment of a local network (if none exists) so that mobile objects can form a network based on the policies of the realm [47,48]. There may be a protocol for communication amongst mobile objects as well as realm managers (See also Generic ID, Section 5). The realm managers should keep track of the existence of each mobile object. The mobile objects can leave or join the network anytime.

As indicated earlier, the realm manager may be responsible for many functions. It is also possible that these functions are split into many small parts, and some servers can take responsibility together with the communication protocol among those servers, such as a location service manager and mapping manager servers.

3.3.9. Signaling and data separation

In IP-based wired/wireless networks, the data and control are in the same communication channel. This may not be true for cellular networks. The separation of control and data in cellular networks introduces many advantages, especially in terms of security. To allow high-speed mobility and optimize the latency, the control signal should be sent over a reliable and fast communication channel.

In addition, the separation of control and data paths can help eliminate the triangulation problem [19] such as in a Mobile IP environment. As in circuit-switched networks, after setting up the data channel, the data can be transferred with minimal overhead.

3.4. Identifier (ID)

In general, there are two types of Identifiers (IDs): a flat ID or a hierarchical ID. Each has its pros and cons. A flat ID may be secure (e.g., a 128-bit user public key), but lacks scalability, and hence may introduce high latency in large systems. The behavior of hierarchical IDs is just the opposite. We recommend using a combination of both, i.e., the flat ID within the domain and hierarchical among domains or realms: *Local ID+Domain ID=Global ID*. Again, the identifier is unique within a domain, and in combination with its domain ID, it is globally unique.

3.4.1. Group ID

A representation of a group of (virtual) objects. This group of objects normally lies within the same domain or realm, but is not restricted to; for example, *channel ID* consists of the layout of many virtual objects. Group ID is also used in multicasting aspects when many (virtual) objects represent the communication end point.

3.4.2. Generic ID

This ID is used for self-organizational purposes. This ID can automatically be assigned to the unknown virtual objects to make them function properly in order to communicate with the rest of the network. Optionally the realm manager may periodically scan the local domain and, in case a new virtual object is present without an identity, this generic ID is assigned.

However, in fact, whenever the virtual object joins the network, the realm manager may announce the presence of the virtual object. Then, based on the realm policies, the realm manager may allow the new virtual object to communicate within/among the domain (s).

3.4.3. Disposable ID

This ID is temporarily created for future applications, and it has an expiration date. Note that the realm manager manages a collection of these IDs, and possibly reuses the unused IDs.

Note that these IDs are just examples. These IDs can apply within a single administrative domain and/or among different domains. These IDs for different object realms/domains can be communicated through inter-realm communication.

3.5. Virtualization architecture challenges

In this section, we describe the impact of the virtualization architecture using ID/locator split concept within the existing routing architecture and the overall system performance.

3.5.1. Routing scalability and architecture

Since the ID/locator split concept is used to separate the functionality of the identity and the locator for each (virtual) object, similar to other ID/locator split approaches [18] the routing scalability issue can be mitigated. In general, this issue is due to the exponential growth of the size of the IP routing table. If a site uses provider aggregatable (PA) addresses, it has to renumber all its hosts when it changes the provider.

On the other hand, it if uses provider independent (PI) addresses, these addresses are not aggregatable, and thus results in an increase of the size of routing tables. With an ID-Locator overlay, it is possible to use PI addresses as Ids and PA addresses as locators. With this approach, only PA addresses are used in the core network, and the scalability issue is resolved. This only occurs when PI is treated as the identity, and PA as the locator. All existing routing mechanisms can function as if it is using the locator, not the identity. However, in the future, since we treat each networkable component as an object, i.e., each (virtual) object may consist of more than one locator, and each is used to establish the channel to form the communication, the future routing architecture may require a modification to allow the virtualization architecture.

3.5.2. System performance

First due to the ID/locator split concept, the complexity of the system ambiguity is reduced; however, this approach increases the levels of mapping, e.g., from name to ID and from ID to locator. Second, due to the virtualization concept, in fact, the system performance may be decreased. For example, consider a virtual host case; one dedicated server obviously performs much better than shared servers; however, the total system performance considering the waste and budget will be improved significantly.

For example, several virtual hosts can operate over a powerful server with high utilization (many to one relationship) rather than one-to-one mapping for each application to a single host. On the other hand, a virtual host can span over multiple low performance servers (one-to-many relationship) as well as a many-to-many relationship. By applying the virtualization architecture, we also simplify the management functionality. It is also possible to move, turn on and off, backup, etc., dynamically without awareness of any physical components. In addition, other key virtualization features, such as fault-tolerance, load balancing/sharing, traffic engineering, etc., can be achieved.

It's also possible that an application provider is not the same as the resource provider (application service vs. hosting service). For example, *Microsoft* may rent a set of virtual servers to provide an application leasing service to the end user. However, again this requires the mechanisms to guarantee the dedicated resource over the shared resources.

3.5.3. Implementation and deployability

Currently, there are some commercial products, e.g., VMware [49], and public domain products, e.g., Xen [50], User Mode Linux [51], and Kernel Based Virtual Machine [52], available which apply the virtualization concept to the server farms, in terms of cloud computing. This is just

one example relating to our architecture; however, our perspective is to scale this example into a large scale network for future Internet usage. Obviously, to achieve our architecture requirements, cooperation amongst the end users, service providers, and software developers, must co-exist. Notice that, in fact, the ID/locator split concept evokes the deployability issue; however, modifying the current Internet (e.g., including many extensions of Mobile IP in order to resolve the mobility problem) has made the Internet more complex and difficult to use, and resulted in additional complex issues.

4. Migrating toward Future Wireless Networks from the current network: examples

In this section, we describe the current network; the networking components applied to those in FWNs, especially how to incrementally migrate the current network into FWNs. Although we have briefly described the current wired/wireless networks in the beginning of this paper, we again revisit some problems and definitions, and then we will focus on the detailed solutions.

4.1. Traditional wired/Wireless and cellular networks: examples

In IP-based wired/wireless networks, one issue is that of renumbering when the network changes its service provider. It may also result in an increase in the number of routing records causing routing scalability problems [6–9]. A site may have multiple interfaces with multiple service providers. One interface is basically used as an egress network. Usually, only one egress is used at a time; the other interfaces are used as backups. Therefore, most of ID/locator split proposals focus on how to achieve *site* multihoming. There is not much discussion on mobility, privacy, etc.

The ID/locator spilt concept is generally proposed so that a provider aggregatable, or PA, address can be used for routing purposes. A provider independent, or PI, address is used as the identity (ID). In the current network, there are normally three configurations.

First, in general, users access the Internet through their corresponding hosts. The hosts do not normally move or change locations, but if moved, it is frequently with a low speed (within the same administrative domain). Host multihoming does not directly affect the performance because the host will usually remain in a single domain. As shown in Table 1, in terms of a TCP/IP reference model, the mapping terminology is as follows: User (User Name) ↔ Host (Host Name, optional TCP/IP port (for TCP-based applications) number, IP address, and MAC address).

Second, in cellular networks, the main focus is on users and mobile devices. In this case, users tend to move across the administrative domain at a high speed. Also, users may own several devices, and each device can consist of many networking interfaces, such as 2G/3G, (Mobile) WiMAX, LTE, Satellite, WLANs, etc. Therefore, the focus is on user/device mobility and user/device multihoming.

With the multihoming feature supported, a user or host is required to maintain and/or utilize multiple interfaces over different networks. For example, a user may have a two-interface device, such as a cellular phone with WiMAX from Clearwire and 3G from T-Mobile. In addition, it is possible that she may also own another mobile device with WLANs and 3G both from T-Mobile. In the first case, the user has two different egress networks, but not in the latter.

The third scenario is when the whole network moves. This movement of the entire network can apply to both wired/wireless and cellular networks. This scenario is similar to the site renumbering in wired/wireless networks, but with a high speed movement.

4.2. Potential virtualization architecture usage: examples

There are many possible applications with the virtualization architecture using the ID/locator split concept. In this section, we provide three examples.

First, currently, mobile devices are more powerful than simply a device for emitting a voice communication but instead provide data-orientated applications. These devices may be compared to a small personal computer in which many applications can run simultaneously. They are mostly equipped with multiple networking interfaces, e.g., 3G, WLAN, WiMAX, and Bluetooth.

One example is to create a virtual mobile device with a virtual networking interface, and so each individual policy can be enforced. Suppose the mobile user wants to run an application requiring different security policies/profiles depending on the domain. With the virtualization architecture using the ID/locator split concept, a single device and/or single or multiple networking interfaces can form the (virtual) channel, and then each individual security policy can be applied directly. There is also an optional interaction amongst these virtual devices. Due to the powerful nature of the device, it can be shared among different people with security and privacy enforcement.

Second, in FWNs, people are more likely to be work-at-home employees. They can basically work anywhere in the world with the network connectivity provided. The travelling cost is substantially reduced, providing people anywhere more opportunities, regardless of the location, and especially people with disabilities. Suppose *Google* provides its employees with this flexibility. One concern is on how to provide the reliable and substantial resource, i.e., bandwidth and delay guaranteed, for the mobile home users so that they can access the information (as if they are working at headquarters). This can be achieved by having the user ID and cooperate ID, with policy enforcement. Suppose the user has two network connections: *Charter* and *AT & T* networks. Traditionally, these service providers do not differentiate the users among other customers. However, *Google* can sign an agreement with both service providers to treat the transaction generated/received by/to its employees with a higher priority.

In fact, this example is similar to when we allow the use of a group ID to access a paid database, e.g., *ieeexplore.ieee.org*. Currently, to access the paid database, authentication is verified by the registered IP addresses,

e.g., University IP addresses. Home users have to access the database through the University proxy server, or over a virtual private network channel. However, with the group ID feature, the home users can directly access the database with the authenticated group ID.

Similarly, suppose *Google* and *Microsoft* provide online application leasing services, such as the database, MS-office, etc. There are two considerations here. First, with the virtualization concept, these companies can place the actual servers in different regions so as to reduce the access latency (this is similar to *Akamai*.) However, all servers are used as a server pool (aka a virtual server). It depends on the requirements and quality of service needed to allocate the resources from this pool to each user/customer.

Note that to make the application leasing feasible, a guaranteed networking resource is also required, which cannot be achieved in the traditional best effort network architecture. In the future, the company may again make an agreement to the service provider to allocate the resources needed for its customer. It is also obvious when Internet is distributed in such a manner (different ownerships, e.g., *Verizon* and *AT* & *T*), a third party, or a service broker, may be required to allocate the resource guaranteed for the entire (virtual) channel, the end-to-end communication.

Third, for our virtualization architecture using the ID/locator split concept, the communication occurs due to a concatenation of (virtual) objects, unlike the end-to-end communication in the current network. In each pair of virtual object to virtual object communication point, a type of messages can be passed over. Compared to the current network, in the future network, we will allow the middle box to be placed anywhere along the (virtual) channel without breaking the end-to-end communication. An example is the proxy. Especially in a mobile wireless environment, the virtualization architecture supports a message passing mechanism; for example, the channel quality which indicates loss/congestion can be sent over to aid the transport protocol to make a transmission decision accordingly.

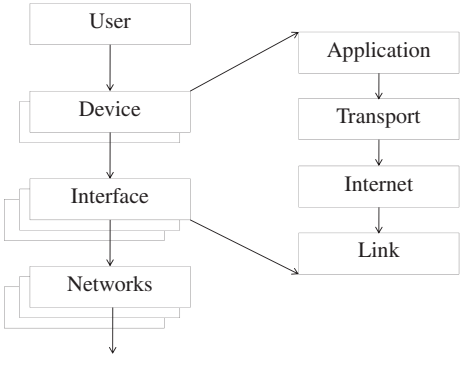

Fig. 5. Simplified Multi-tier Object Model: User, Device + Interface (Application, Transport, Internet, and Link), and Networks.

Note that, similar to a system call and an application programming interface, the virtualization architecture generalizes enough to provide the interface between the user and the virtualized networking architecture. End users can query and submit the requirement, and it is up to the architecture to determine whether it can fulfil the requirement, with corresponding costs. The network architecture is open, and this allows the user to develop new services using the provided interface or even a new interface.

4.3. Preliminary steps toward FWNs

The united ubiquitous communication protocol is expected to be IP-based in FWNs. A traditional TCP/IP protocol stack potentially will be applied to FWNs. In current networks using a virtual object concept, we can apply the virtualization using the ID/locator split concept in that each physical/logical component is treated as an individual object. Each object has an identity (ID) and optional locators and names.

For simplicity, initially there will be no virtual objects (no virtual devices or virtual interfaces). There are four tier abstractions-each physical and logical: user, device, interface, and network; and application, transport, Internet, and link as shown in Fig. 5. Tables 2 and 3 provide examples of the mapping tier for wired/wireless and cellular networks used in FWNs.

4.4. Identification (ID) toward FWNs

Each object has an individual ID: user ID, device/node/host ID, interface ID, network ID (e.g., router/switch ID) as well as application ID, transport ID, internet ID, and link ID. The ID consists of flat and hierarchical portions. These IDs are unique within a particular domain, the local ID. A hash of the public key can be used as a simple representation of the unique ID for security.

Table 3
Logical mapping-tier: Application, Transport, Internet, and Link.

Protocol Stack	Name	ID	Locator
Application	DNS User Name	Application ID	N/A
Transport	N/A	TCP Port Number	N/A
Internet	N/A	IP address	N/A
Link	N/A	MAC address	N/A

Table 2
Physical Mapping-Tier (3-tiers): User, Device or Host, and Interface.

Tier	Wired/Wireless Networks			Cellular Networks		
	Name	ID	Locator	Name	ID	Locator
User	DNS User Name	N/A	IP or MAC address	User Name	User ID	Telephone Number
Device or Host	DNS Host Name	IP/MAC address	IP or MAC address	Device Name	IMSI and/or IMEI	Telephone Number
Interface	N/A	IP/MAC address	IP or MAC address	N/A	N/A	N/A

Table 4
Physical Mapping-Tier for current networks applied for FWNs Example: User, Device or Host, and Interface.

Tier	Name	ID	Locator
User	User Name	Hash (IMEI or user identification or public key)	Based on Device/Host
Device or Host	Device/Host Name	Hash (IMSI or serial number. or public key)	Based on Interface
Interface	Interface Name	MAC address	IP address or Telephone Number

A combination of the local ID and its domain serves as a global ID. We recommend an identity representation of 128 bits to allow backward-compatibility with legacy nodes (in IPv6 systems). The division of the number of bits for local and global IDs is arbitrary. It may be better to use multiples of 32 bits, particularly if there is a transaction or computation involving 32-bit IPv4 addresses.

Table 4 shows an example of a modification in the current mapping-tier. Note that to simply make a current TCP/IP model support mobility when the host moves, TCP/IP IDs can be used as an indirection level, so that the network connection is formed, and tied to this ID, not the IP address or locator shown in Fig. 6.

4.5. Mapping systems

In wired/wireless networks, there are two levels of mapping from a FQDN DNS name to an IP address (by DNS resolution) then to a MAC address (by ARP resolution). In cellular networks, usually only one level is required to map from a user name to telephone number. Therefore, in FWNs, we reorganize the mapping explicitly and specify the function for each term. Unlike in traditional networks, we consider both inter-tier and intra-tier resolution.

In addition, we consider a vertical and horizontal resolution. From Name to ID or from ID to locator, we call this horizontal resolution, and between IDs, called vertical resolution. For some static bindings, DNS-like systems may be used. For more dynamic bindings, such as ID to locators and vertical resolutions, extra mapping severs will be required.

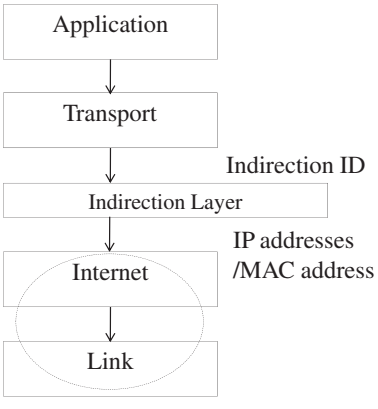

Fig. 6. Logical Mapping-Tier for current networks applied in FWNs Example: Application, Transport, Indirection Layer, Internet, and Link.

4.5.1. Mapping examples

Table 5 shows a configuration of a mapping example from the current network to FWNs. In this scenario, for simplicity, there is no organization or domain relationship (the identity is secure flat ID.) The flat ID makes it difficult to enforce policies, and it may also lead to scalability issues.

In this example, John Smith uses a service which implicitly states that John Smith will be subscribed into a particular domain. A simple example is that John Smith is working at Washington University in St. Louis, and he registers one of his locators, i.e., WLANs to the University network. His ID may contain a hierarchical portion of his domain, such as country code (USA) and school (Washington University in St. Louis).

Therefore, in order to reach John Smith, his ID will result within the Washington University in St. Louis mapping server. Then, this organization can enforce its policies on the appropriate communication channel. John Smith may also have a global flat identifier stored at the US realm manager server. Then, the US realm manager server will redirect all transactions to the Washington University in St. Louis realm server, if this is the mapping registered at the US server.

4.5.1.1. Mobility. Whenever John Smith moves from one place to another; his ID remains the same, but not his locators. Note that this movement within the same domain may not require any extra procedures except an update of IDs and locators. However, if John Smith moves to different domains, an agreement among service providers is required to allow roaming by John Smith with policy enforcement.

4.5.1.2. Multihoming. Since John Smith owns many devices with different interfaces, he may probably register several locators to the mapping servers. Suppose an IP address represents a locator, the ID/locator split concept allows the

Table 5
Mapping example.

Case study	User, John Smith, owns Iphone with (3G and WLAN) interfaces; and his laptop with Ethernet and WLAN interfaces.
User:	User name: John Smith and his ID, Hash (John Smith Public Key).
Device:	Device names are Iphone 3Gs and IBM laptop. Device IDs are Hash (IMSI of Iphone) and Hash (IBM serial number).
Interface:	No explicit interface name; however, WLANs, 3G, WiMAX can be used on interfaces names. Interfaces IDs are MAC addresses and 3G identifications. At this level, IP addresses and/or telephone numbers can represent the *locations* of these interfaces.

mapping from an identity to many different locators with different/same (load sharing/load balancing) weights. To achieve the desired load sharing/balancing, cooperation amongst his service providers and/or his organizational policy, especially for the ingress transaction, is required.

4.5.1.3. User location privacy. Since John's User ID is used to reach him, and not his exact location; the user location privacy may be maintained. However, the hierarchical part of his ID may implicitly reveal his location in terms of his organization; in this case, there is a trade-off amongst the management/scalability and privacy. Note that the border router and/or proxy servers can also operate as a source address rewritten functionality in order for truly hiding the domain ID from each user.

5. Experimental study

In this section, we demonstrate the feasibility and proof of concepts of our virtualization architecture using the ID/locator split concept. There are two setups.

First, we want to show the feasibility by applying the virtualization concept to segregate/constrain shared resources among virtual objects (e.g., virtual hosts and virtual networking interfaces), and second, to apply the ID/Location Split concepts to our virtualization architecture.

Note that in our testbed, we simplified several networkable components and made the best resource usage of our available hardware and software. In our setup, the virtual objects are limited to only the virtual hosts and virtual networking interfaces. We also restricted the corresponding bandwidth to each virtual networking interface so that the virtual network can support the maximum bandwidth requirement. We used a rendezvous server as our simplified realm server. This server primarily does the mapping, resource management, and quality control functions.

In general, in our testbed (Table 6), there is one server (Dell Precision Workstation 690) with two CPUs (Intel Xeon) 3 GHz each, two 1-gigabit Ethernet interfaces, 2 GB memory, and a 150 GB SATA (Serial Advanced Technology Attachment) disk. This server provides virtual host functionality. There are four virtual hosts installed, each sharing only 1 CPU, 256 MB virtual memory, and 8 GB virtual disk. We chose the virtualization software, VMware ESXI version 4.0 [49], to implement the virtualization architecture. Each virtual host has Linux Ubuntu 9.10 (kernel version 2.6-31-19) installed as the operating system.

Table 6
Hardware and software configurations.

Virtualization Server	Dell Precision Workstation 690 (Intel Xeon 3 GHz) 2 GB memory 2 × Intel Gigabit Ethernet Western Digital 150 GB SATA disk
Forwarding DHCP Server	Intel Pentium 4 (2 GHz) 512 MB memory Seagate 80 GB IDE disk 2 × 100 Mbps Ethernet
Software	VMware ESXI version 4.0 OpenHIP Linux Ubuntu 9.10 (Kernel 2.6-31-19)

We chose OpenHIP version 0.7 (running in a user mode) [53] as our representation of the ID/locator split concept installing it into each virtual host.

Each networking interface attaches to an individual Ethernet switch (to separate the broadcast network). In addition, there is another server providing the forwarding mechanism between two networks, using the basic Linux IP forwarding mechanism. This server also provides DHCP (Dynamic Host Configuration Protocol) functionality. This server is Intel Pentium 4 running at 2 GHz with 512 MB memory, 80 GB IDE (Integrated Drive Electronics) disk, and two 100 Mbps Ethernet interfaces. The operating system is Linux Ubuntu 9.10 with Linux kernel version 2.6.31-19. Note that in each setup, we conducted five trials for our experiment.

5.1. Experimental configurations

There are two configurations. Fig. 7 shows the first configuration in managing the virtualization architecture. In this figure, at the server machine, we installed two virtual hosts (acting as clients, called *vid2* and *vid3*); each virtual host attaches to the virtual networking interface at 1 Mbps. For server functionality, we setup another virtual host, called *vid1*, with 100 Mbps as a virtual networking interface.

Virtual channels are setup between the server and each client. There are also two 100 Mbps Ethernet switches to separate the broadcast networks. We ran an *iperf* program (version 2.04) [54] to simulate the traffic using UDP (User Datagram Protocol) for 10 Mbps bandwidth running over 20 s. The traffic was generated from a server to two clients in different UDP ports (different channels).

Fig. 8 shows the second configuration by applying the ID/locator split concept into the virtualization architecture. In this setup, there are only two virtual hosts running on the server. Each virtual host attaches to 100 Mbps virtual networking interfaces in different networks: 192.168.1.x and 192.168.2.x. The average delay between the two virtual hosts is in a range between 1 to 3 ms.

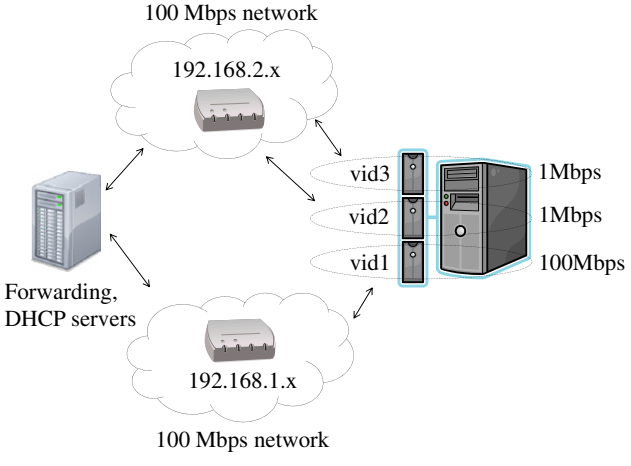

Fig. 7. Network setup I (virtualization architecture).

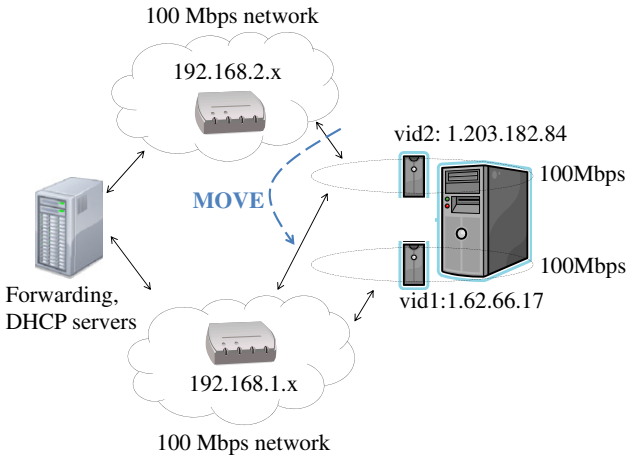

Fig. 8. Network Setup II (ID/Locator Split).

Then, we installed OpenHIP in each virtual host. After installation, the first virtual host, or *vid1*, has an IP address 192.168.1.5 and 192.168.2.11 for the other, or *vid2*. The local HIT (Host Identity Tag), or LSI [53], for each host is 1.62.66.17 and 1.203.182.84, respectively. Note that *vid1* also acted as a rendezvous server (our realm server) for mapping purposes. Also, the resource management, one of the realm functionalities, was controlled by a VMware server.

We ran a *ping* command to check the reachability from *vid1* to *vid2*. We also ran a *Secure Shell* application built-in Linux Ubuntu; we then logged into *vid1* from *vid2*. An example of the ping command is shown in Table 7. Then, we moved the second virtual host, *vid2*, to another network (the IP address is also changed, not the LSI), and observed the results. Note that we measured the overall hand-off timing from when the link was first disconnected until the result from the ping command continued.

5.2. Experimental results and discussions

Based on the two configurations, as shown in Figs. 7 and 8, the results show that the achieved bandwidth for *vid2* and *vid3* is 947 kbps and 968 kbps, respectively. This behavior implies that the virtualization architecture can segregate the resource, i.e., bandwidth, among different virtual networking interfaces.

Note that for shared resource aggregation purposes, it is also possible to generate multiple virtual networking interfaces within a single virtual host (based on the virtualization of software features), and this virtual host can benefit from these multiple virtual networking interfaces.

Table 7
Results from a ping command between two virtual hosts.

```
vid2 > ping 1.62.66.17
...
64 bytes from 1.62.66.17: icmp_seq = 21 ttl = 63 time = 2.35 ms
64 bytes from 1.62.66.17: icmp_seq = 22 ttl = 63 time = 1.27 ms
...
```

For example, in this testbed, we also created two virtual networking interfaces for an individual virtual host, and thus, one virtual host has two different unique IP addresses. Notice that the way to utilize the resource, i.e., load sharing, over multiple interfaces is out of the scope of this paper. Normally to increase the bandwidth capacity, a flow distribution and/or bandwidth aggregation technique can be applied [5,55–57]. The purpose of this experimental study is to show only the flexibility of resource sharing for our virtualization architecture.

Second, as shown in Fig. 8 (with a ping command), the communication was established between two virtual hosts, based on the host identity, not the IP address. In addition, during the hand-off, i.e., changing the network of the second virtual host (*vid2*), it took around 13 seconds of physical movement from one network to another (unplugging and plugging the physical networking interface; until the virtual host binds the new IP address to its corresponding interface), and also requires around 24 s on average for this virtual host (*vid2*) to continue the operation (Secure Shell login) without a disconnection. In other words, mobility can be maintained if, for example, the IP address is changed, but not the host identity.

Notice that we also did the experiment in a non-OpenHIP scenario, and the results showed that the connection was frozen; the secure shell login operation could not continue.

In this testbed, OpenHIP and VMware are just two of the software-representations for our concept. These do not represent all features for our virtualization architecture using the ID/locator split concept; the purpose of this testbed is to show the feasibility of the concept. So, the performance will solely depend on the software/hardware implementation, i.e., using other platforms may achieve better or worse performance. However, this performance comparison is not our focus. As described earlier, we simplified many features and networking components. For Future Wireless Networks, many objects and virtual objects require further investigation.

6. Related work

Several ID/locator split approaches [18] have been proposed to resolve both mobility and multihoming problems. In these proposals, the functions of the identity and the locator are explicitly separated.

In general, the splitting is based on the indirection concept. Internet Indirection Infrastructure [19], or*i3*, is one of the first such indirection concepts. *i3* was abstractly introduced as the trigger concept on overlay networks. The idea, briefly, is as follows: senders transmit the packet using unique host IDs.

The network abstractly forwards the injected packets to the node whose ID is matched, by using look-up mechanisms similar to those in peer-to-peer, or P2P, services. In *i3* overlay networks, servers store the trigger and forward packets between end points. Therefore, *i3* requires the assumption that the end host knows the lists of *i3* servers.

Many proposals have been introduced and have been derived from the indirection concept into the ID/locator

split context. The ID/locator split concept has been applied to wired/wireless networks, in that a host has its own unique identity. When the host moves, its identity remains unchanged, but not its locators. The identity may be a string of characters or digits. The locator is only a representation of the current point of attachment to the networks.

In packet-switched networks, the locator is used to decide where the packet should be routed to. In circuit-switched networks, cellular networks in particular, the mobile phone number is used as the identity, and the roaming server provides the locator; therefore, the mobile phone number remains the same regardless of the location.

In the current IP architecture, each host, or node, has a name and an address. The host name is Fully Qualified Domain Name (FQDN). The IP address represents both an identity and a locator. Again, this intermixture makes it difficult to achieve full mobility, multihoming, and privacy, as we have already discussed. The domain name server (DNS) is used to convert from FQDN to the host IP address. Then, the same address is used as a locator for routing the packet to the end host.

In a wired/wireless network, there are three ways to implement the ID/locator split concept: split at the end host (e.g., Mobile IP [20,21], NEMO [58], Virtual ID [4], HIP [59], and SHIM6 [60]), split in the network (LISP [61]), and use a hybrid of both approaches (e.g., Enhanced MILSA [62,63], HRA [64], and SIX/ONE [65]).

The first approach requires the use of a tunneling (encapsulation) or the insertion of a new ID sub-layer between the transport and network layers. Thus, the upper layers are bound to the host ID instead of its locators. The second set of splitting techniques implements the ID/locator split concept in the network. The basic advantage is that there is no change to the end hosts; the routers take care of the split. At the edge of the network, IDs are resolved into the locators needed for the actual communication. This requires changes to the network infrastructure devices, e.g., routers. The third approach is to combine the former two and allow the splitting in both the host and the network, with a complexity trade-off.

Enhanced MILSA introduces a new secure namespace. SHIM6 uses one of its current locators as the identity. Therefore, SHIM6 does not support mobility of the end host. Mobile IP and Virtual ID use a home address and a virtual home address as the identity. Mobile IP, Virtual ID, and SHIM6 may be easier to deploy since a new naming space is not required; a traditional hierarchical IP address structure can still be applied. However, a permanent home address or virtual home address is necessary. NEMO is based on Mobile IP, but a mobile router (MR) does the mobility function on behalf of its mobile nodes.

In general, this splitting technique can support full mobility, multihoming, and location privacy, because the identity is used instead of the host location. Many indirection mechanisms require new naming spaces and additional Name/ID/Locator resolution mechanisms. However, none of these techniques can achieve all key requirements for FWNs.

In addition, most ID/locator split proposals have no ownership representation, and so that makes it difficult to apply any policy enforcement. Internet 3.0 [15] and PONA [22] introduce the concept of objects in three tiers: user, host, and location. The policy enforcement is achieved in each tier via the use of realm (or domain) managers.

Observe that for all approaches we have described, each individual technique cannot achieve all key requirements for FWNs. As a result, with the inspiration of ID/Locator Split [18], Objects [66], Policy Enforcement (both trusted and untrusted domains) [22], and ID Privacy [4], we combine these four approaches and present a new framework and/or architecture to achieve the FWN key requirements.

In this architecture, we generalize both physical and logical representations of networkable components into the so-called *objects*[15,66], and also derive the concept of end-to-end communication, called a *channel*. We propose a virtualization concept of the object and its communication, called *virtual object*, to form a multi-tier architecture solution [17]. These multi-tier concepts are based on an ID layer perspective.

Especially in cellular networks, for example, users' IDs allow users to have several networking devices (user locators), device IDs allow devices to support multiple networking interfaces (device locators), and interface IDs allow each interface to have multiple points of attachment (interface locators). In this example, the resolution occurs in a three-tier mapping.

For the current wired/wireless network, the Internet, a two-tier architecture may be adequate: users to networking devices and devices to networking interfaces. Applying the ID/locator split concept, an IP address is bound to each interface and only used as the locator. In addition, we derive the identity from a combination of the hierarchical and secure flat domains, e.g., a DNS system and a hash of public key, in order to mitigate the scalability issue of the new namespace.

In addition, the concept of realm, a virtual trust domain, is introduced for relationships and ownership purposes. A realm manager is used for policy enforcement, mobility tracking and decision making of multi-interface connectivity, etc. Unlike other encapsulation techniques such as Mobile IP, we separate the control and data paths in order to reduce the delay latency and encapsulation overhead.

Similar to SIX/ONE, we provide an option for mobile user objects to choose the networking connectivity; however, it also allows network service providers to recommend the optimal connectivity. Thus, a border router or a realm manager in our proposed FWNs acts as a proxy in data plane and performs an optional address rewriting mechanism. With this address rewriting technique, mobile user objects' location privacy can be maintained. In addition, the provider aggregatable, or PA address, can be deployed for routing scalability purposes.

7. Conclusions

Future Wireless Networks (FWNs) shall be a cloud interconnected via IP-based core. This united ubiquitous networking protocol needs to operate with many different wired/wireless/cellular technologies with various types of future applications. There are many features that FWNs

should support and provide, such as interoperability, guaranteed service, scalability, mobility, multihoming, path preference selection, privacy, security, deployability, etc.

With a traditional IP architecture, one of the greatest obstacles to achieve full mobility and multihoming is the overloading of IP addresses used as both identity and locator. As a result, in this paper, we focus on this particular problem and propose the separation implied by these two functions.

In this paper, aside from the ID/locator split proposals, compared to other existing proposals on this separation, the architecture we recommended here is limited to just the current networking components, such as hosts, routers, and networking interfaces. We generalized the splitting concept of networkable components as *objects*. We also applied the virtualization concept into these components. Therefore, our architecture is generalized for FWNs.

In addition, we predict the future service as a virtualization of all components so that we again apply this concept to each end-to-end communication, *channel*, with individual policy enforcement. Therefore, it is easy to support full virtual object mobility and multihoming. In addition, the policies can be applied to each virtual object and channel (end-to-end communication). Our framework also allows the concept of object ownership to achieve resource sharing/leasing.

Finally, we demonstrated the feasibility and proof of concepts for our virtualization architecture using the ID/locator split concept by showing a simplified model in terms of resource sharing and mobility supports.

References

[1] IEEE, IEEE Standard for Local and Metropolitan Area Networks, Part 21: Media Independent Handover Services, IEEE STD 802.21-2008, 2009.

[2] M. Blanchet, Ed. P. Seite, Multiple Interfaces Problem Statement, Internet-Draft, draft-blanchet-mif-problem-statement-02.txt, 2010.

[3] M. Wasserman, Ed., Current Practices for Multiple Interface Hosts, Internet-Draft, draft-ietf-mif-current-practices-00, 2009.

[4] C. So-In, R. Jain, J. Pan, S. Paul, Virtual ID: A Technique for Mobility, Multihoming, and Location Privacy in Next Generation Wireless Networks, in Proceedings of IEEE Consumer Communication and Networking Conference, 2010, pp. 1–5.

[5] C. So-In, R. Jain, S. Paul, J. Pan, A Policy Oriented Multi-Interface Selection Framework for Mobile IPv6 Using the ID/Locator Split Concepts in the Next Generation Wireless Networks, in the 2nd Int. Conf. on Computer and Automation Engineering (ICCAE), Feb. 2010.

[6] D. Raychaudhuri, M. Gerla, Ed., New Architectures and Disruptive Technologies for the Future Internet: The Wireless, Mobile and Sensor Network Perspective, NSF Wireless Mobile Planning Group Workshop, Aug. 2005.

[7] L. Kleinrock, A vision for the Internet, ST J. for Research 2 (1) (2005) 4–5.

[8] D. Clark, R. Braden, K. Sollins, J. Wroclawski, D. Katabi, New Arch: Future Generation Internet Architecture, Technical Report, Air Force Research Laboratory, 2003, 76 pp.

[9] A. Perrig, D. Clark, S. Bellovin, Ed., Secure Next Generation Internet, NSF Workshop Report, 2005.

[10] DARPA Internet Program, Internet Protocol, RFC 791, 1981.

[11] S. Deering and R. Hinden, Internet Protocol Version 6 (IPv6) Specification, RFC 2460, 1998.

[12] The 3rd Generation Partnership Project, The 3GPP Technical Specification Group Service and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access, 3GPP TS 23.401 V8.0.0, 2007, 167 pp.

[13] The 3rd Generation Partnership Project, The 3GPP Technical Specification Group Service and System Aspects; Architecture enhancements for non-3GPP accesses, 3GPP TS 23.402 V8.0.0, 2007, 131 pp.

[14] ITU-T, General requirements for ID/locator separation in NGN, Y.2015, 2007, 18 pp.

[15] R. Jain, Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation, in: Proceedings of IEEE Military Communication Conference, 2006, pp. 1–9.

[16] D. Meyer, L. Zhang, K. Fall, Report from the IAB Workshop on Routing and Addressing, RFC 4984, 2007.

[17] S. Paul, J. Pan, R. Jain, A Survey of Naming Systems: Classification and Analysis of the Current Schemes Using a New Naming Reference Model, WUSTL Technical Report, 2009. http://www.cse.wustl.edu/~jain/papers/naming.htm.

[18] C. So-In, R. Jain, S. Paul, Future Wireless Networks: Key Issues and a Survey (ID/Locator Split Perspective), WUSTL Technical Report, 2009. http://www.cse.wustl.edu/~jain/papers/fwns.htm.

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

[20] C. Perkins, Ed., IP Mobility Support for IPv4, RFC 3220, 2002.

[21] D. Johonson, C. Perkins, J. Arkko, Mobility Support in IPv6, RFC 3775, 2004.

[22] S. Paul, R. Jain, J. Pan, M. Bowman, A Vision of the Next Generation Internet: A Policy Oriented Perspective, in: Proceedings of British Computer Society International Conference on Visions of Computer Science, 2008.

[23] L. Fang, W. Du, P. Ning, A beacon-less location discovery scheme for wireless sensor networks, in: Proceedings of IEEE Conference on Computer Communication, 2005, pp. 161–171.

[24] J. Kuo, W. Liao, Modeling the behavior of flooding on target location discovery in mobile ad hoc networks, in: Proceedings of IEEE International Conference on Communication, 2005, pp. 3015–3019.

[25] S.S. Yau, W. Gao, D. Huang, A Location-based Directional Route Discovery (LDRD) Protocol in Mobile Ad-hoc Networks, in: Proceedings of Global Telecommuncaitons Conference, 2006, pp. 1–6.

[26] X. Shao, L. Heng-Ngoh, T. Kiong-Lee, T. Yoong-Chai, L. Zhou, J.C.M. Teo, Multipath cross-layer service discovery (MCSD) for mobile ad hoc networks, in: Proceedings of Services Computing Conference, 2009, pp. 408–413.

[27] C.A. Perryea, S. Chung, Community-Based Service Discovery, in: Proceedings of International Conference on Web Services, 2006, pp. 903–906.

[28] P. Patel, S. Chaudhary, Context Aware Semantic Service Discovery, in: Proceedings of World Conference on Serivces, 2009, pp. 1–8.

[29] D. Zhu, M.W. Mutka, C. Zhiwei, Using cooperative multiple paths to reduce file download latency in cellular data networks, in: Proceedings of Global Telecommuncaitons Conference, 2005, pp. 2480–2484.

[30] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, B. Patil, Proxy Mobile IPv6, RFC 5213, 2008.

[31] K. Shiomoto, I. Inoue, E. Oki, Network virtualization in high-speed huge-bandwidth optical circuit switching network, in: Proceedings of IEEE Conference on Computer Communication, 2008, pp. 1–6.

[32] P. Garbacki, V.K. Naik, Efficient Resource Virtualization and Sharing Strategies for Heterogeneous Grid Environments, in: Proceedings of IFIP/IEEE International Symposium on Integrated Network Management, 2007, pp. 40–49.

[33] T. Hossfeld, K. Leibnitz, A. Nakao, Modeling of Modern Router Architectures Supporting Network Virtualization, in: Proceedings of Global Telecommuncaitons Conference, 2009, pp. 1–6.

[34] Y. Wei, J. Wang, C. Wang, Bandwidth Guaranteed Multi-Path Routing as a Service over a Virtual Network, in: Proceedings of International Conference on Inteligent Networks and Intelligent Systems, 2008, pp. 221–224.

[35] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, RSVP-TE: Extensions to RSVP for LSP Tunnels, RFC 3209, 2001.

[36] J. Marko, P. Mikko, R. Jukka, Lease-Based Resource Management in Smart Spaces, in: Proceedings of Pervasive Computing and Communication Workshops, 2007, pp. 622–626.

[37] B. Sotomayor, R.S. Montero, I.M. Llorente, I. Foster, Resource Leasing and the Art of Suspending Virtual Machines, in: Proceedings of High Performance Computing and Communication, 2009, pp. 59–68.

[38] G. Urdaneta, G. Pierre, M.V. Steen, A Survey of DHT Security Techniques, IEEE ACM Computing Surveys (2009).

[39] H. Soliman, Mobile IPv6 support for dual stack Hosts and Routers (DSMIPv6), Internet-Draft, draft-ietf-mip6-nemo-v4traversal-05.txt, 2007.

[40] T. Kalibera, F. Pizlo, A.L. Hosking, J. Vitek, Scheduling Hard Real-Time Garbage Collection, in: Proceedings of Real-Time Systems Symposium, 2009, pp. 81–92.
[41] V.V. Kapadia, D.G. Thakore, Distributed Garbage Collection Using Client Server Approach in Train Algorithm, in: Proceedings of Advance Computing Conference, 2009, pp. 492–495.
[42] T. Liu, Z. Ma, Z. Ou, A Novel Process Migration Method for MPI Applications, in: Proceedings of the 15th IEEE Pacific Rim International Symposium on Dependable Computing, 2009, pp. 247–251.
[43] T. Maoz, A. Barak, L. Amar, Combining Virtual Machine migration with process migration for HPC on multi-clusters and Grids, in: Proceedings of IEEE International Conference on Cluster Computing, 2008, pp. 89–98.
[44] Y. Akai, K. Wakao, T. Yokouchi, M. Kai, Development of the strong migration mobile agent system AgentSphere for autonomic distributed processing, in: Proceedings of IEEE Pacific Rim Conference on Communication, Computers and Signal Processing, 2009, pp. 582–587.
[45] W. Kuang-Lai, C. Shieh, C. Hsu, Service migration–a new paradigm for content distribution systems, in: Proceedings of the 3rd International Conference on Communication and Networking in China, 2008, pp. 34–38.
[46] J. Park, J. Kim, H. Choi, Y. Woo, Virtual machine migration in self-managing virtualized server environments, in: Proceedings of International Conference on Advanced Communication Technology, 2009, pp. 2077–2083.
[47] N. Wakamiya, S. Arakawa, M. Murata, Self-Organization Based Network Architecture for New Generation Networks, in: Proceedings of the 1st International Conference on Emerging Network Intelligence, 2009, pp. 61–68.
[48] M.O. Cherif, S.M. Senouci, B. Ducourthial, A new framework of self-organization of vehicular networks, in: Proceedings Global Information Infrastructure Symposium, 2009, pp. 1–6.
[49] VMware ESXI v.4, 2010. http://www.vmware.com/products/esxi.
[50] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauery, I. Pratt, A. Warfield, Xen and the Art of Virtualization, in: Proceedings of the 22nd ACM Symposium on Operating Systems Principles, 2003, pp. 164–177.
[51] The User-mode Linux Kernel. http://user-mode-linux.sourceforge.net/.
[52] Kernel Based Virtual Machine. http://www.linux-kvm.org/.
[53] OpenHIP 2009. http://www.openhip.org.
[54] Iperf. http://sourceforge.net/projects/iperf.
[55] H. Hsieh, R. Sivakumar, A Transport Layer Approach for Achieving Aggregate Bandwidths on Multi-Homed Mobile Hosts, in: Proceedings of International Conference on Mobile Computing and Networking, 2002, pp. 83–94.
[56] H. Adiseshu, G. Parlkar, G. Varghese, A Reliable and Scalable Striping Protocol, ACM SIGCOMM Computer Communication Review 26 (4) (1996).
[57] J.C. Fernandez, T. Taleb, M. Guizani, N. Kato, Bandwidth aggregation-aware dynamic QoS negotiation for real-time video streaming in next-generation wireless networks, IEEE Transactions on Multimedia 11 (6) (2009) 1082–1093.
[58] K. Nagami, S. Uda, N. Ogashiwa, H. Esaki, U. Tokyo, R. Wakikawa, H. Ohnishi, Multihoming for Small-Scale Fixed Networks Using Mobile IP and Network Mobility (NEMO), RFC 4908, 2007.
[59] R. Moskowitz, P. Nikander, P. Jokela, Host Identity Protocol (HIP) Architecture, RFC 4423, 2006.
[60] E. Nordmark, M. Bagnulo, Internet Draft: Shim6: level 3 multihoming Shim protocol for IPv6, Internet-Draft, draft-ietf-shim6-proto-09, 2007.
[61] D. Meyer, The Locator Identifier Separaton Protocol (LISP), Cisco Systems: The Internet Protocol J. 11 (1) (2008).
[62] J. Pan, S. Paul, R. Jain, M. Bowman, MILSA: A Mobility and Multihoming Supporting Identifier Locator Split Architecture for Naming in the Next Generation Internet, in: Proceedings of IEEE Global Communication Conference, 2008, pp. 1–6.
[63] J. Pan, S. Paul, R. Jain, M. Bowman, S. Chen, Enhanced MILSA Architecture for Naming, Addressing, Routing and Security Issues in the Next Generation Internet, in: Proceedings of IEEE Global Communication Conference, 2009, pp. 1–6.
[64] X. Xu, D. Guo, Hierarchical routing architecture, in: Proceedings of the 4th Euro-NGI Conference on Next Generation Internetworks, 2008, pp. 7
[65] C. Vogt, Six/One: a solution for routing and addressing, in: IPv6, Internet-Draft, draft-vogt-rrg-six-one-01.txt, 2007.
[66] G. Booch, Object-Oriented Design with Applications, The Benjamin Cummings Publishing Company, Inc., CA, 1991.

Chakchai So-In received his B.E. and M.E. degrees in Computer Engineering from Kasetsart University, Bangkok, Thailand in 1999 and 2001, respectively. In 2003, He was an internetworking trainee in a CNAP program (sponsored by Cisco Systems) at Nanyang Technological University (NTU), Singapore and also obtained Cisco Career Certifications, e.g., CCNP and CCDP. He also received M.S. and Ph.D. in Computer Engineering from the Department of Computer Science and Engineering, Washington University in St. Louis (WUSTL), MO USA in 2006 and 2010, respectively. In summer 2006, He was a student intern at mobile IP division, Cisco Systems, CA USA. He was also a student intern at WiMAX Forum and Bell Labs during summer 2008 and 2010. His research interests include architectures for Future Wireless Networks/Next Generation Wireless Networks and Future Internet; congestion control in high speed networks; protocols to support network and transport mobility, multihoming, and privacy; and quality of services (QoSs) in (broadband) wireless access networks, i.e., (Mobile) WiMAX and LTE (Advanced).

Raj Jain is a Fellow of IEEE, a Fellow of ACM, a winner of ACM SIGCOMM Test of Time award and ranks among the top 50 in Citeseer's list of Most Cited Authors in Computer Science. Dr. Jain is currently a Professor of Computer Science and Engineering at Washington University in St. Louis. Previously, he was one of the Co-founders of Nayna Networks, Inc-a next generation telecommunications systems company in San Jose, CA. He was a Senior Consulting Engineer at Digital Equipment Corporation in Littleton, Mass and then a professor of Computer and Information Sciences at Ohio State University in Columbus, Ohio. He is the author of "Art of Computer Systems Performance Analysis," which won the 1991 "Best-Advanced How-to Book, Systems" award from Computer Press Association. His fourth book entitled "High-Performance TCP/IP: Concepts, Issues, and Solutions," was published by Prentice Hall in November 2003.

Subharthi Paul received his BS degree from University of Delhi, Delhi, India, and Master's degree in Software Engineering from Jadavpur University, Kolkata, India. He is presently a doctoral student in Computer Science and Engineering at Washington University in St. Louis, MO USA. His primary research interests are in the area of Future Internet Architectures.

Jianli Pan received his B.E. in 2001 from Nanjing University in Posts and Telecommunications (NUPT), and M.S. in 2004 from the Beijing University of Posts and Telecommunications (BUPT), China. He is currently a Ph.D. student in the Department of Computer Science and Engineering in Washington University in Saint Louis, MO USA. His current research is on the next generation Internet architecture and related issues. He is currently a student member of the IEEE.