Wireless Netw (2008) 14: 71–85
DOI 10.1007/s11276-006-7818-1

HISNs: Distributed gateways for application-level integration of heterogeneous wireless networks*

Phone Lin · Huan-Ming Chang · Yuguang Fang · Shin-Ming Cheng

Published online: 9 June 2006

Abstract Integration of different kinds of wireless networks to provide people seamless and continuous network access services is a major issue in the B3G network. In this paper, we propose and implement a novel Heterogeneous network Integration Support Node design (HISN) and a distributed HISN network architecture for the integration of heterogeneous networks, under which the Session Mobility, Personal Mobility, and Terminal Mobility for mobile users can be maintained through the Session Management mechanism. Thus, the HISN node can serve as an agent for the user to access Internet services independent of underlying communication infrastructure. Our design is transparent to the bearer networks and the deployment of the HISN network does not need to involve the operators of the heterogeneous wireless networks.

Keywords Heterogeneous networks · Heterogeneous network integration support node (HISN) · Personal mobility · Service mobility · Session mobility.

*This paper is an extension of the work that won the championship of the Mobile Hero contest sponsored by Industrial Development Bureau of Ministry of Economic Affairs, Taiwan, R.O.C., and was awarded USD 30,000. The work of Lin, Chang and Cheng was supported in part by the National Science Council (NSC), R.O.C, under the contract number NSC94-2213-E-002-083 and NSC94-2213-E-002-090, and NSC 94-2627-E-002-001, Ministry of Economic Affairs (MOEA), R.O.C., under contract number 93-EC-17-A-05-S1-0017, the Computer and Communications Researches Labs/Industrial Technology Research Institute (CCL/ITRL), Chunghwa Telecom Labs, Telcordia Applied Research Center, Taiwan Network Information Center (TWNIC), and Microsoft Corporation, Taiwan. The work of Fang was supported in part by the US National Science Foundation Faculty Early Career Development Award under grant ANI-0093241 and US National Science Foundation under grant DBI-0529012.

P. Lin
P. Lin, Department of Computer Science & Information Engineering and Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 106, R.O.C.
e-mail: plin@csie.ntu.edu.tw

H.-M. Chang
Department of Computer Science & Information Engineering, National Taiwan University, R.O.C.
e-mail: r91114@csie.ntu.edu.tw

Y. Fang (✉)
Department of Electrical & Computer Engineering, University of Florida, USA
e-mail: fang@ece.ufl.edu

S.-M. Cheng
Department of Computer Science & Information Engineering, National Taiwan University, R.O.C.
e-mail: shimi@pcs.csie.ntu.edu.tw

1. Introduction

In recent years, technologies for wireless networks such as the terrestrial cellular systems, wireless local area networks (WLANs), and wireless personal area networks (WPANs) have been evolved quickly. The huge evolution, together with the advances in computing capability of mobile devices, enable us to use various kinds of devices to access Internet services. Integration of different kinds of wireless networks (also known as heterogeneous networks) to provide us seamless and continuous network access services becomes a major issue in the B3G or 4G networks.

Several studies [4,13,14] have been conducted to address the issues for the integration of heterogeneous networks. First, protocol and message format conversion for different network protocols and device capabilities should be designed. Second, while users roam among different networks, the network should maintain the continuity for the active

sessions, which is known as the "Session Mobility". Third, "Personal Mobility" to users residing in different kinds of networks should meet two requirements:

Requirement 1. Reachability, that is, the user data can be routed to the mobile device the user currently uses.

Requirement 2. Personalization, i.e., a personalized operating environment (e.g., storage space or preference setups for users' applications) can be provided by the networks.

Fourth, the existing network protocols should not be notified with the new infrastructures introduced to integrate the heterogeneous networks. Moreover, due to the exposure of communication media and information exchanges, security issue should be considered. Besides the above issues, the so called "Terminal Mobility" should also be addressed, which is defined as that within the coverage area of a network, the network is able to track the locations of users' terminals, and seamlessly provide transmission services for the users. For the Terminal Mobility management, protocols exist for most of current wireless networks. Examples are GPRS Mobility Management [9], Mobile IP [6], or Session Initiation Protocol [7]. To provide the terminal mobility in the heterogeneous networks, we may reuse the existing terminal mobility mechanisms run in the different wireless networks.

The previous studies [4,13–15] have developed platforms for the integration of heterogeneous wireless networks. The Mobile People Architecture (MPA) project [14,15] aimed to achieve the Reachability requirement of Personal Mobility, which is a logical extension of the current networking model and targets to place the users, rather than the devices, at the endpoints of a communication session. A new layer, called *personal layer*, is introduced on top of the application layer. This layer focuses on personal-level routing, which allows messages to be routed to users' current network and device with content conversion. The MPA uses the globally unique Personal Online ID to identify a user, and utilizes the personal proxy located at the user's home network for location tracking. The personal proxy will receive communication services on behalf of the user and forward it to the user. This can hide the user's real location, achieving location privacy. The personal proxy provides the necessary communication services such as email, telephones, and ICQ messages, and being independent of networks makes the personal proxy easy to extend. Through the introduction of person-routing in MPA, the MPA lacks, however, the personal operating environment, which is also an important part of personal mobility.

The ICEBERG project [12,13] at U.C. Berkeley integrates the cellular telephony networks with Internet to provide the Reachability requirement of Personal Mobility. The same as MPA, the ICEBERG treats a user as a communication endpoint regardless of the device he/she uses. It is based on Nina clustering computing platform [21], which takes care of call setup and control, location tracking, and mapping user names in heterogeneous networks into ICEBERG naming. The ICEBERG network is an overlay network of iPOPs on top of the Internet. However, the use of ICEBERG access point (which are bridges between various networks and ICEBERG) needs modification of existing network components, such as switches or bases stations in Public Switched Telephone Network (PSTN), which is a difficult task for practical deployment. Besides, the ICEBERG emphasizes more on reachability rather than personal operating environment.

The IPMoA [4] is a mobile-agent-based personal mobility framework that utilizes mobile agents to support two requirements of personal mobility. There are three kinds of agents, including Personal Application Assistant (PAA), Personal File Assistant (PFA), and Personal Communication Assistant (PCA). The PAA invokes applications and routes program data for users. The PFA provides the access to user's personal files and maintains the synchronization for the files in the different IPMoAs. The PCA handles the call establishment and content conversion. When the user is located on a visiting network of IPMoA, the PAA invokes the remote methods to remotely execute applications in home network.

In these previous works, the Session Mobility issue was not addressed. Most works adopted a centralized node architecture to provide Personal Mobility for mobile users, in which The centralized node may become a bottleneck. Besides, most of them do not follow the standard to come out with the security functionality, which significantly decreases the integrity with Internet.

Besides the above previous studies, the 3GPP working group proposes the Virtual Home Environment concept (VHE) [2] to provide seamless personal mobility services, including personalized services and personal operating environment, on the Universal Mobile Telecommunications System (UMTS) for end users. However, the VHE approach limits only on UMTS networks and devices, which may not be applicable to other kinds of heterogeneous networks.

The Wireless Application Protocol (WAP) forum proposes the WAP gateway [25] to connect the mobile cellular network with the Internet, which adopts the proxy approach. The WAP gateway targets on the translation between the WAP and the HTTP protocols, which does not maintain the personal or session mobility for the users.

In this paper, we propose the design of a *Heterogeneous network Integration Support Node* (HISN) for the integration of heterogeneous wireless networks. Then, we study a distributed HISN network architecture under which seamless integration of heterogeneous networks is made possible. Under this architecture, the HISN network maintains the Session Mobility, Personal Mobility, and Terminal Mobility for mobile users through the Session Management mechanism while an HISN serves as an intelligent agent for mobile users in Internet. Through the HISN network, the mobile users

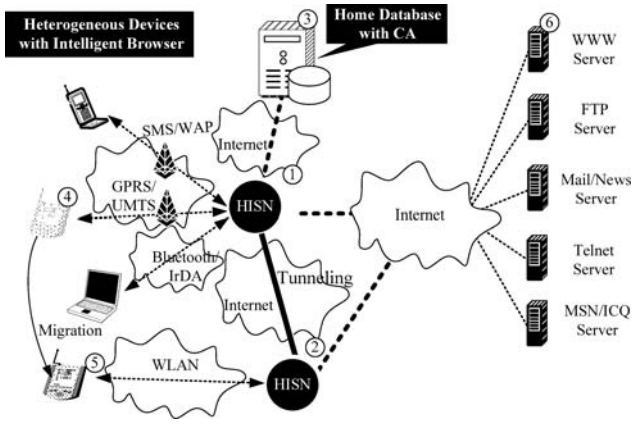


Fig. 1 The distributed HISN network architecture

can use different kinds of mobile devices to access Internet service. To address the security over HISN network, we deploy a standard security protocol "PKI" [5] for the secure communication between two HISNs.

The rest of this paper is organized as follows. Section 2 illustrates the distributed HISN network architecture. Section 3 describes the implementation of the HISN. Section 4 details the procedures for the Session Management mechanism in this network. Section 5 evaluates the performance of the HISN network. We conculde this paper in Section 6.

2. The HISN network architecture

This section discusses the HISN network architecture. As shown in Fig. 1, there are three kinds of nodes in the HISN network: HISN nodes (Figs. 1 (1) and (2)), Home Database with Certification Authority (Fig. 1 (3)), and Intelligent Browsers (IBs; Figs. 1 (4) and (5)). The HISN node is a gateway for users, through which users may use different kinds of devices through different kinds of networks to access Internet application servers (Fig. 1 (6)).

In the HISN network, the user may connect to the HISN node through two kinds of connection media: the public-domain media (e.g., GPRS or UMTS) and the private-domain media (e.g., WLAN, IrDA, and Bluetooth). For an HISN node, there may be more than one public-domain media or private-domain media connected. The user may roam between the service areas of different HISNs. To make the user gain seamless service between an old node and a new HISN node, the Session Management mechanism is deployed (to be elaborated later) in the HISN node. The functionalities of each node in the HISN network are described as follows.

HISN Node: See Figs. 1 (1) and (2). The HISN node is the primary serving node in the HISN network. In a HISN node, we implement the Session Management mechanism to support personal mobility, session mobility, and terminal mobility for mobile users. The security mechanism is accommodated in the HISN architecture to guarantee the secure access to the HISN network. In an HISN node, the connection-media interfaces for heterogeneous networks (e.g., IrDA, WLAN, Bluetooth, and GPRS) are provided to allow users to use different mobile devices to access the HISN. We implement a content format translation function to encapsulate the content obtained from Internet with the format that can be displayed on the mobile terminal. The protocols (e.g., Telnet, FTP, and E-Mail) of the Internet applications are implemented in an HISN. The details for the implementation of an HISN node will be elaborated in Section 3.

UserID	Password	ServingHISN
User A	abcd	IP of HISN B
User B	1234	--
User C	wxyz	IP of HISN X

Fig. 2 Examples of entries in home database

Home Database with Certificate Authority: See Fig. 1 (3). This database maintains the service information for the users. When a user subscribes to the HISN network, a permanent entry is created for the user. The entry consists of three fields: UserID (to store the ID[1] of the user), Password (to store the password for authorization of the user), and ServingHISN (to store the IP address of the HISN where the user currently resides at). Figure 2 shows examples of the entries in the Home Database.

Following the PKI[2] infrastructure [5], the Certificate Authority in the Home Database node is responsible for authenticating and publishing certificates to the mobile devices and the HISN nodes involved. With the certificates, the mobile devices and HISNs can be verified for their validity, and the communication information (between a mobile device and an HISN or between two HISNs) can be encrypted/decrypted by the public and private secure key.

Intelligent Browser (IB): See Figs. 1 (4) and (5). An IB is a software component installed in the user's device to communicate with an HISN node. We implement an intelligent user interface on IBs.

Due to the page limitation, this paper omits the details of the implementation for the Home Database with CA and the

[1] The UserID is a unique ID in the HISN network for the mobile user to identify himself/herself.

[2] PKI is a system of public key encryption using digital certificates from Certificate Authorities and other registration authorities that verify and authenticate the validity of each party involved in an electronic transaction.

IB. We focus on the implementation of an HISN node and the Session Management mechanism.

3. The implementation of an HISN node

Figure 3 illustrates the software architecture of an HISN, which consists of five components: Connection Media Adapter, Content Format Adapter, Management Part, Personal Operating Environment, and Internet Service Module. The details of these components are given below:

Connection Media Adapter: See Fig. 3 (1). This component is responsible for sending/receiving the user data to/from mobile devices transparently through heterogeneous networks. In a heterogeneous network, this component acts as a terminal, and an HISN can be addressed by the ID (e.g., IP address or phone number) used in the heterogeneous network. Thus, the deployment of an HISN need not involve the operators of individual networks. Currently, an HISN can connect to five kinds of networks: Infrared [24], Bluetooth [23], Ethernet, WLAN [16], and GPRS [9]. The data are delivered to these different network interfaces directly through the comm port or TCP connection. Following our design, an HISN node can be easily extended to connect to other kinds of networks as time evolves.

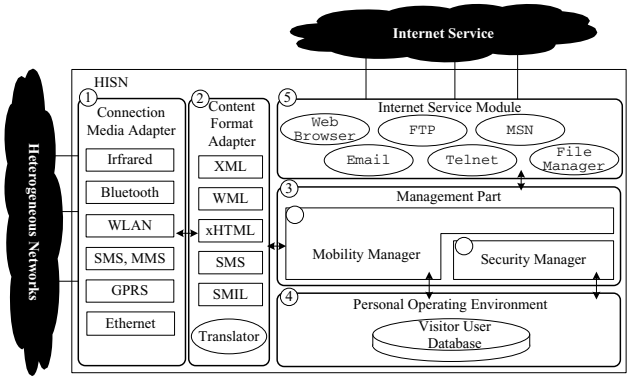
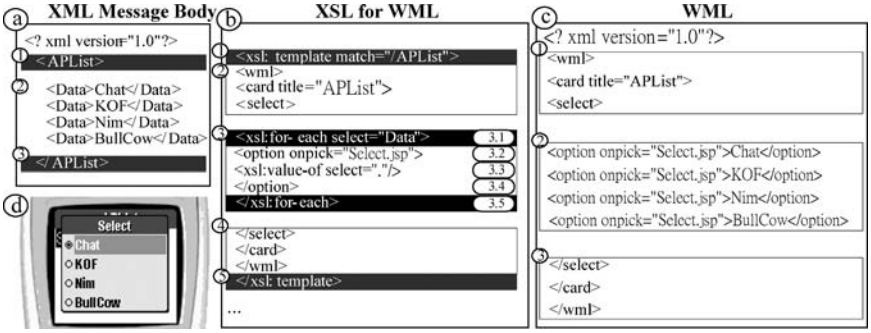


Fig. 3 The software architecture of an HISN

Content Format Adapter: See Fig. 3 (2). This component handles the content format translation. According to the mobile device a user currently uses, this component dynamically encapsulates the content (received from the Internet or a mobile device) into the format that can be interpreted by and shown on the mobile device. We implement the Translator to convert the content into different formats, where the eXtensible Stylesheet Language (XSL) [17] is used to express the translation rule. Currently, five content formats are supported by an HISN: the eXtensible Markup Language (XML) [1], the Wireless Markup Language (WML) [25], the Extensible Hypertext Markup Language (xHTML) [8], the Short Message Service (SMS) [18], and the Synchronized Multimedia Integration Language (SMIL) [10]. Figure 4 demonstrates how the Translator translates an XML message to a WML message. The content between the ⟨xsl:template match="/APList"⟩ and ⟨/xsl:template⟩ tags (Figs. 4 (b.1) and (b.5)) describes the translation rule for the block of data between the ⟨APList⟩ and ⟨/APList⟩ tags (Figs. 4 (a.1) and (a.3)). The content between the ⟨xsl:for-each select="Data"⟩ and ⟨/xsl:for-each⟩ tags (Figs. 4 (b.3.1) and (b.3.5)) describes the translation rule between the ⟨Data⟩ and ⟨/Data⟩ tags (Fig. 4 (a.2)). According to the script defined in Figs. 4 (b.3.2)–(b.3.4), the content between two ⟨Data⟩ tags is removed and encapsulated into a WML message format (c.2), and the (b.2) and (b.4) blocks are added to the head, (c.2), and the end, (c.3), respectively.

Management Part: See Fig. 3 (3). This part consists of two managers: the Security Manager and the Mobility Manager. The Mobility Manager (Fig. 3 (3.1)) routes the messages and data between a mobile device and an Internet Service Module (to be elaborated later; Fig. 3 (5)). When the mobile user roams among different HISNs, he/she may activate different applications on the different HISNs. Suppose that the user currently resides in HISN **A** and activates an application. The Mobility Manager in HISN **A** stores the application information, the IP address of HISN **A**, in the Visitor

Fig. 4 An example for the translation from the contents in XML format into that in WML format



User Database in the Personal Operating Environment (to be elaborated later; Fig. 3 (4)). The Mobility Manager also stores the user-related information (including the address of the mobile device, e.g., IP address or MSISDN, and the IP address of HISN **A**) in the Visitor User Database. This information is referenced for routing. The Mobility Manager accommodates the Session Management mechanism to correctly route the user data. The details of the Session Management mechanism is described in the next section. The Security Manager (Fig. 3 (3.2)) is responsible for authentication of communication entities, authorization of the legal usage of resource, and security of the user data. The implementation of the Security Manager follows the Public Key Infrastructure (PKI) standard, X.509 [5].

Personal Operating Environment: See Fig. 3 (4). This component maintains the user-related information. There is a Visitor User Database in the Personal Operating Environment, which maintains the mobile user information. When a mobile user accesses an HISN node, a UserProfile context is created in the Visitor User Database for the user. As shown in Table 1, a UserProfile context consists of five fields: UserID, CurrentHISN, DeviceProfile, NetworkProfile, and UserState. The UserID stores the User ID. The CurrentHISN field stores the IP address of the HISN which the user currently accesses. The DeviceProfile field stores the related-information (i.e., device type; e.g., PDA or notebook) of the mobile device the user currently uses. The NetworkProfile field keeps the related-information (i.e., the network type and the MSISDN or IP address of the mobile device) for the network (the user currently accesses). The UserState maintains the status for the HISN network access of the user, which consists of five states: IDLE, LOGIN, RESUMING, ACTIVE, or STANDBY. A corresponding state machine (to be elaborated later) is run for the transitions of the five states.

When a mobile user activates a session through an HISN node to the Internet, a Session context is created to maintain the status of the session, which is contained in the user's UserProfile context. Each UserProfile context may contain zero or more Session contexts. As shown in Table 1, a Session context consists of seven fields: SessionID, LocatedHISN, ApplicationName, SessionState, TunnelID, SecretKey, and BufferedData. The SessionID field stores the ID of the session, which is used as the reference key for searching the Session context. The LocatedHISN field stores the IP address of the HISN where the user creates the session. The ApplicationName field stores the types (e.g., Telnet or Email) of the application for the session. The SessionState field maintains the status of a session: **ACTIVE, SUSPEND**, and **MIGRATE**. We maintains a state machine for the transitions of the three states, which will be described later. The TunnelID stores the ID of the tunnel serving the session. The SecretKey field stores a symmetric ciphering key to encrypt/decrypt the transmitted user data. The BufferedData stores the file pointer to the file used to buffer the user data of a session.

Internet Service Module: See Fig. 3 (5). This part implements the classes of the agents for users to access Internet applications, e.g., FTP and Telnet. Currently, we implement six kinds of agents, `Web Browser`, `FTP`, `MSN`, `Email`, `Telnet`, and `File Manager`. Our implementation can be easily extended to accommodate more application agents. The `Web Browser` agent runs the HTTP protocol to retrieve the web information in the Internet. The `FTP client` agent provides functions for downloading/uploading files from/to an Internet host. The `MSN`,`Email`, and

Table 1 HISN UserProfile and session contexts

Field	Description
UserID:	The primary reference key to search the UserProfile context.
CurrentHISN:	The IP address of the HISN the user currently accesses.
DeviceProfile:	Related-information of the device the user currently uses.
NetworkProfile:	Related-information of the network through which the user accesses the HISN.
UserState:	User state: IDLE, LOGIN, RESUMING, ACTIVE, or STANDBY.
Each UserProfile context may contain zero or more of the following Session contexts:	
SessionID:	The identity of the application session.
LocatedHISN:	The IP address of the HISN where the session is created.
ApplicationName:	The application type of the session.
SessionState:	Session state: ACTIVE, SUSPEND, or MIGRATE.
TunnelID:	The identity of the tunnel through which the user data is delivered.
SecretKey:	The ciphering key of the secured tunnel.
BufferedData:	The file pointer to the file used to buffer the user data.

`Telnet` agents serve as the clients for the instant messaging, telnet, and Email applications, respectively. The `File Manager` agent is a client software implementing functions to access the Internet remote user file storage, where the Network File System (NFS), Server Message Block (SMB), and Common Internet FileSystem (CIFS) protocols are implemented.

4. The session management mechanism

This section describes the Session Management mechanism in the HISN network, which consists of seven procedures: the Login procedure, the Application Activation procedure, the Application Termination procedure, the Logout procedure, the Suspend procedure, the Resume procedure, and the Tunneling procedure. Before the user gains the service of a HISN node, the Login procedure is executed between the mobile device and an HISN node. After the execution of the Login procedure, the user can activate the Internet application through the HISN node by executing the Application Activation procedure. The user can turn off the running Internet applications through the Application Termination procedure. The Logout procedure is exercised when the user quits the service of the HISN node. During the execution of an application, the mobile user may move out of the service area of an HISN and move into the service area of another HISN node. At this moment, the Suspend procedure is exercised between the old HISN node and the mobile device, and the Resume procedure is executed between the new HISN node and the mobile device to continue the application session. Then, the Tunneling procedure is run between the old HISN node and the new HISN node for the packet forwarding from the old HISN node to the new HISN node. The details of the seven procedures are given in the following few subsections.

4.1. State machines for session management

The Session Management maintains a finite state machine S_u for the user behavior and a finite state machine S_s for session status. The state diagrams for S_u and S_s are shown in Figs. 5 and 6, respectively.

The S_u is maintained in the UserState field of the UserProfile context, and characterized by one of the five different states: **IDLE**, **LOGIN**, **RESUMING**, **ACTIVE**, and **STANDBY**. At the **IDLE** state, the UserProfile context holds invalid information for user's location, device, and network. The user is unreachable in this case. The Login procedure and authentication are executed at the **LOGIN** state while the UserProfile context is not valid. At the **RESUMING** state, the Resume procedure is performed, where the Session contexts can be updated or resumed. At this time, the UserProfile context is valid but the Session contexts (if any) are invalid. At the **ACTIVE** state, the user may perform the Application Activation and/or Termination procedures. The corresponding Session contexts will be created or terminated. In this state, the user is reachable by the HISN network and the UserProfile and Session contexts are valid. At the **STANDBY** state, the Suspend procedure is performed, where the messages and user data are buffered.

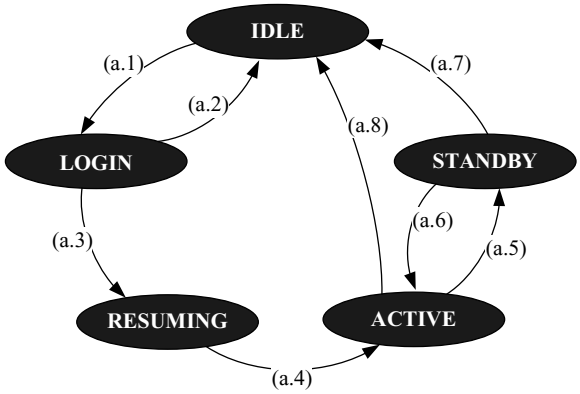


Fig. 5 State diagram for state machine S_u for user behavior

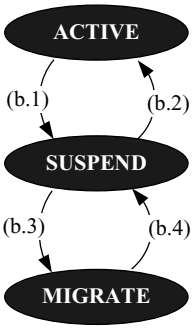


Fig. 6 State diagram for state machine S_s for session status

The S_s is maintained in the SessionState field of the SessionProfile context, and characterized by one of the three different states: **ACTIVE**, **SUSPEND**, and **MIGRATE**. At the **ACTIVE** state, the session is activated through the Application Activation procedure. The application can deliver the user data at this state. At the **SUSPEND** state, the activated session is suspended and the HISN node will buffer messages and maintain the application session for the user. At the **MIGRATE** state, the user performs the Resume procedure among different HISN nodes, the TunnelID and SecretKey fields are updated and the messages or user data are delivered through the tunnel between two different HISN nodes.

The state transitions in the S_u and S_s state machines are described in the following subsections.

4.2. The login procedure

Suppose that a user previously accessed the HISN node H_o through the access network N_o, and then he/she logins into the HISN node H_n, where the access network is N_n. Figure 7

Table 2 An example message encapsulated with the inner message format

Field	Example	Description
Message:	APPLICATION_DATA	Message name
Sender:	User	Source of the message
Receiver:	InternetServiceModule	Destination of the message
UserID:	User A	Identity of the user
Args:	FTP	Application type
Args:	S1	Session ID, identity to the application session
Args:	GET	The parameter for the application
Args:	'a.mpg'	The parameter for the application
...	...	...

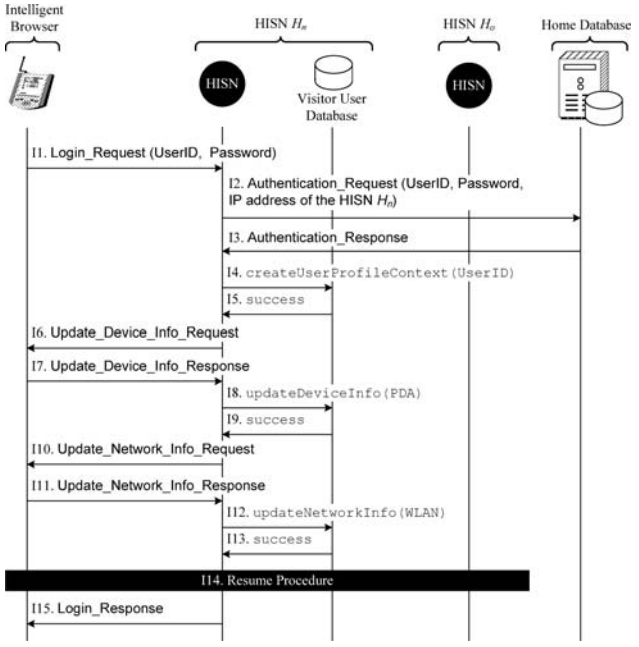


Fig. 7 The message flow for the Login procedure

illustrates the message flow for the Login procedure with details given below. All the messages to be proceed is encapsulated with the inner message format. Table 2 shows the details of the format, which consists of four major fields: Message, Sender, Receiver, and UserID. The Message field specifies the command in the message. The Sender field specifies which component (e.g., Mobility Manager or Internet Service Module) issues the message, and the Receiver field identifies which component will receive the message. The UserID field stores the ID of the user. Besides the four major fields, an inner message may contain variable number of arguments stored in the Args field.

Step I1. The mobile device establishes a connection to the H_n node, which can be done by layer one and layer two of H_n's access network. Here, we use the Bluetooth access network [23] as an example to illustrate the establishment of the connection (For other kinds of access networks, the establishment procedures are similar). The mobile device activates the Bluetooth Discovery service to check whether any Bluetooth connection is available. The mobile device uses the serial port profile broadcasted by H_n to establish the connection. After that, the data can be exchanged between the mobile device and the HISN node. The IB sends a Login_Request(UserID, Password) message to H_n. Note that the transmission between user's device and the HISN node can be secured through the inherent security mechanism in the bearer network, for example, the Wired Equivalent Privacy (WEP) protocol in the IEEE 802.11 network, the secure link layer in the Bluetooth network, and A3, A8, and A5 algorithms in the GPRS network. To reduce the implementation complexity, our design utilize these security mechanisms.

Step I2. Upon receipt of the Login_Request message, the Content Format Adapter encapsulates Login_Request with the inner message format. The Mobility Manager in H_n sends an Authentication_Request message to the Home Database, which contains the user's ID, password, and the IP address of H_n. Note that in our design, we implement the PKI protocol between an HISN node and the Home Database. The password is encrypted following the PKI protocol for secure transmission.

Step I3. After receiving the Authentication_Request, the Home Database authenticates the user by comparing the UserID and the Password. If the authentication succeeds, the Home Database updates the ServingHISN field as the IP address of H_n. Then the Home Database replies the HISN with an Authentication_Response message, which contains the authentication result and the IP address of H_o. Note that if the user previously did not login any HISN node, the returned ServingHISN field is left empty.

Steps I4 and I5. The Mobility Manager in H_n invokes the `createUserProfileContext(UserID)` function to create a UserProfile context for the user in the Visitor User Database. The default value of UserState is **IDLE**. The IP address of H_n is filled into the CurrentHISN field. The user's network address

obtained from the Connection Media Adapter is stored in the NetworkProfile field. Then, the Mobility Manager changes S_u from **IDLE** to **LOGIN** (Transition a.1 in Fig. 5). The Visitor User Database returns a `success` value to the Mobility Manager.

Step I6. The Mobility Manager in H_n sends IB a Update_Device_Info_Request message to request the configuration information of the mobile device.

Step I7. The IB responds the Update_Device_Info_Resonse message to the Mobility Manager of H_n to update the information about the type of the mobile device (e.g., PDA). The detection of the capability of the mobile device (e.g., CPU speed, memory size, or screen size) is done through the invocation of the standard API provided by the OS of a mobile device [22].

Steps I8 and I9. The Mobility Manager invokes the `updateDeviceInfo()` function to store the information for the mobile device type to the DeviceProfile field of user's UserProfile context in the Visitor User Database. The Visitor User Database returns the `success` value to the Mobility Manager.

Steps I10 and I11. The H_n and the IB exchange the Update_Network_Info_Request and Update_Network_Info_Response message pair to update the information about the access network, e.g., GPRS or WLAN.

Steps I12 and I13. The Mobility Manager of H_n invokes the `updateNetworkInfo()` function to update the NetworkProfile field in the UserProfile context for the user in the Visitor User Database. Then the Visitor User Database returns a `success` value to the Mobility Manager.

Step I14. The H_n triggers the Resume procedure to resume the sessions that were created in the H_o node. The details of the Resume procedure will be elaborated later.

Step I15. The HISN H_n sends a Login_Response message to the IB to indicate the result of the Login procedure. After successful login, the user may access the Internet applications through the H_n node.

4.3. Application activation procedure

This section describes the Application Activation procedure for a user to create a session. We use the FTP application as an example. For other applications, the message exchanges are similar, which are not presented in this paper. Suppose that a user activates an FTP on the HISN H_o. Figure 8 illustrates the message flow for the Application Activation procedure with details given below.

Step A1. The IB sends an Activate_Application_Request(ApplicationName: FTP) message to H_o.

Steps A2 and A3. Upon receipt of the Activate_Application_Request message, the Internet Service Module activates the FTP application and an FTP client by creating an object of the FTP client to serve the user. A session ID (e.g., S1) is generated for the session. The Mobility Manager invokes the `createSessionContext()` function to create a corresponding Session context in the Visitor User Database, where the SessionID, LocatedHISN, and ApplicationName fields are filled with S1, the IP address of H_o, and FTP, respectively. The state for the session is set to **ACTIVE**. The TunnelID and SecretKey fields are left as blanks. A file is opened to buffer the data for the session, whose name is filled into the BufferedData field. Then, the Visitor User Database returns a `success` value to the Mobility Manager of H_n.

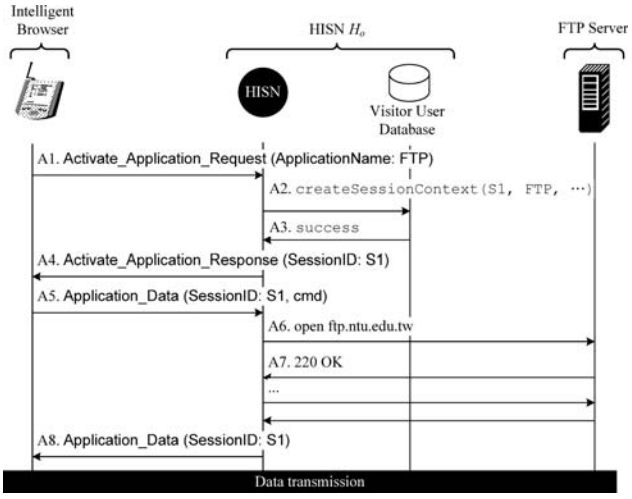


Fig. 8 The message flow for the application activation procedure

Step A4. An Activate_Application_Response message is sent to the IB, where the session ID, S1, is carried in this message.

After Step A4, the agent in the Internet Service Module is ready to serve for the mobile device. In the following steps (i.e., Steps A5–A8), we show how the IB of the mobile device instructs the agent to access the FTP service.

Step A5. Before downloading/uploading files from/to the remote FTP server, the user commands the FTP client to connect to the remote FTP server by sending an Application_Data (SessionID: S1, cmd) message to H_o, where the session ID and the commands are carried in this message.

Steps A6 and A7. Upon receipt of the Application_Data message, the agent (i.e., the FTP client) follows the standard FTP protocol [20] to create a connection between the H_o and the FTP server in the Internet. Then, the IB can communicate with the FTP server through H_o.

Step A8. The H_o sends the Application_Data (SessionID: S1) message to the IB, which contains the execution results for the user.

4.4. Suspend and resume procedures

By performing the Suspend and Resume procedures, the user can move out of the service area of an HISN and enter the service area of another HISN without loss of an ongoing session.

The range of the service area of an HISN is determined by the connection medium through which the user accesses the HISN (i.e., the service area of the HISN is equal to the radio coverage of the connection medium by which the user accesses the HISN). For example, if the user uses the Bluetooth to access the HISN, the service area of the HISN is equal to the radio coverage of the Bluetooth.

Assume that the user currently uses the device D_o to connect to the HISN H_o through the Network N_o, and later he/she uses the device D_n to connect to the HISN H_n through the Network N_n. Seven possible scenarios can be considered for the user movement:

Case I: *Inter-device movement.* $D_o \neq D_n$, $N_o = N_n$, and $H_o = H_n$. For example, the user may change the mobile device for better mobility (e.g., from a desktop to a PDA).

Case II: *Inter-connection medium movement.* $D_o = D_n$, $N_o \neq N_n$, and $H_o = H_n$. As mentioned previously, there are two kinds of connection media: the public-domain media and the private-domain media. The user may choose the private-domain media instead of the public-domain media for the economical concern, security reasons, connection availability, and the capabilities of connection media.

Case III: *Inter-HISN movement.* $D_o = D_n$, $N_o = N_n$, and $H_o \neq H_n$. The user may roam from one HISN to another due to the limited range of the HISN's service area or other considerations as mentioned in Case II.

Case IV: *Inter-device and connection medium movement.* $D_o \neq D_n$, $N_o \neq N_n$, and $H_o = H_n$. In this case, the user changes the mobile device and the type of the access network for the same HISN.

Case V: *Inter-device and HISN movement.* $D_o \neq D_n$, $N_o = N_n$, and $H_o \neq H_n$. The user uses different mobile devices among different HISNs, but does not change the access network to the HISN.

Case VI: *Inter-connection medium and HISN movement.* $D_o = D_n$, $N_o \neq N_n$, and $H_o \neq H_n$. The user uses the same device to access different HISNs through different access networks.

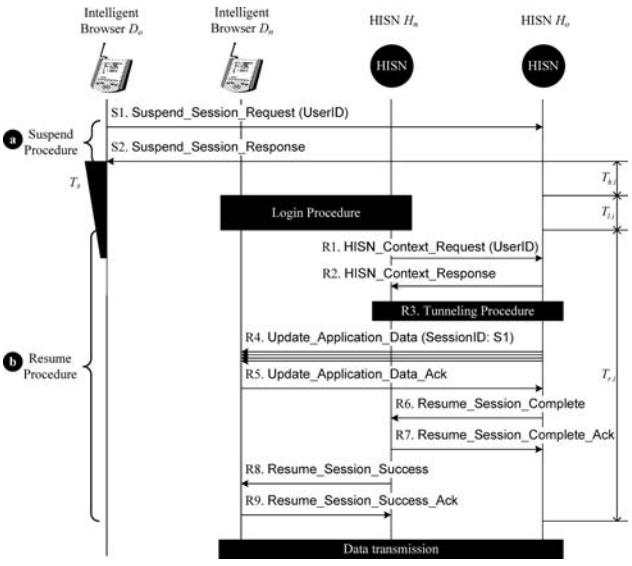


Fig. 9 The message flow for the suspend and resume procedures

Case VII: *Inter-device, connection medium, and HISN movement.* $D_o \neq D_n$, $N_o \neq N_n$, and $H_o \neq H_n$. When a user roams to a new HISN H_n, he/she uses different device and different type of network to access H_n.

The Suspend procedure is executed between the old HISN H_o and the mobile device before the user moves out of the service area of H_o. The HISN network suspends the activated sessions and buffer the data for the sessions. When the user logins into the new HISN node H_n, the Resume procedure is executed to continue the suspended application.

In this subsection, we illustrate the Suspend and Resume procedures by considering Case VII where $D_o \neq D_n$, $N_o \neq N_n$, and $H_o \neq H_n$. For other cases, the message flows for the two procedures can be easily modified. As shown in Fig. 9 (a), suppose that the user suspends the application with the session ID, S1, at the old HISN H_o, where the user uses the mobile device, D_o. Then the user logins into the new HISN H_n by using a new mobile device, D_n.

Suspend Procedure:

Step S1. The IB sends a Suspend_Session_Request(UserID) message to H_o, where the user ID is carried in this message.

Step S2. Upon receipt of Suspend_Session_Request, the Mobility Manager of H_o changes the state of S_s from **ACTIVE** to **SUSPEND** (Transition b.1 in Fig. 6), and starts to buffer the data for the S1 session into the file whose name is specified in the BufferedData field of the Session context. The state of S_u is changed from **ACTIVE** to **STANDBY** (Transition a.5 in Fig. 5). Then, H_o responds a Suspend_Session_Response message to D_o, and starts a timer, T_s. The H_o expects to receive an

HISN_Context_Request message (to be elaborated later; see Step R2) from other HISN node before the T_s timer expires. When the T_s timer expires, the Suspend procedure is exited.

After entering the service area of H_n, the user executes the Login procedure by using the new mobile device D_n. During the execution of the Login procedure, the H_n determines the previous HISN (i.e., H_o) by querying the Home Database (see Step I2.2). Then the Resume procedure is executed among D_n, H_n, and H_o. Figure 9 (b) illustrates the message flow for the Resume procedure with details given below.

Resume Procedure:

Step R1. The Mobility Manager of H_n sends a HISN_Context_Request(UserID) message with the user's ID to the Mobility Manager of H_o to request the Session contexts of the user(if any). Note that the IP address of the HISN H_o is obtained in Step I2.2 in the Login procedure. Then, the Mobility Manager changes the state of S_u from **LOGIN** to **RESUMING** (Transition a.1 in Fig. 5).

Step R2. Upon receipt of HISN_Context_Request, the Mobility Manager of H_o retrieves the Session context of the user by querying the Visitor User Database where the user's ID is used as the reference key. The Mobility Manager of H_o sends the HISN_Context_Response message to the Mobility Manager of H_n, which contains the Session contexts for the user (i.e., Session context for S1).

Step R3. After receiving the HISN_Context_Response message, H_n stores the received Session context into the user's UserProfile context in the Visitor User Database. Then the Mobility Manager checks the LocateHISN fields in the Session context to see whether any other HISN maintains suspended sessions for the user. If so, the H_n executes the Tunneling procedure to establish the tunnels between H_n and the previous HISNs. Details of the Tunneling procedure will be given in the next subsection.

Step R4. After Step R3, the tunnel between H_n and H_o has been established. If the H_o has buffered data (for the sessions that are created in H_o) to be delivered to D_n, H_o sends one or more Update_Application_Data (SessionID: S1) messages carrying the data to D_n through the tunnel.

Step R5. After receiving the buffered data, D_n sends a Update_Application_Data_Ack message to H_o through H_n for acknowledgment. The Mobility Managers of H_n and H_o change the states of their S_s state machines from **SUSPEND** to **MIGRATE** (Transition b.3 in Fig. 6), indicating that the data can be exchanged through the tunnel.

Note that if there are more than one session to be resumed, Steps R4 and R5 are executed repeatedly until all the sessions in the HISN H_o are resumed.

Steps R6 and R7. The H_o sends H_n a Resume_Session_Complete message, which indicates that all the sessions on H_o are resumed. Then H_n returns a Resume_Session_Complete_Ack message. The Mobility Manager of H_o changes the state machine S_u from **STANDBY** to **ACTIVE** (Transition a.6 in Fig. 5) for the user.

If there are other HISNs having activated sessions for the user, H_n will take the same actions in Step R3 to create one tunnel, dedicated for the user, for one HISN, and Steps R4-R7 are executed repeatedly to resume the activated sessions on H_n for the user.

Steps R8 and R9. The H_n and IB D_n exchange the Resume_Session_Success and Resume_Session_Success_Ack messages pair. The Mobility Manager of H_n changes the state of S_u from **RESUMING** to **ACTIVE** (Transition a.4 in Fig. 5). For the data of the applications that are created in H_o, they are delivered among D_n, H_n, H_o, and the application servers.

4.5. Tunneling procedure

When a mobile user activates applications on different HISNs, tunnels will be established between the HISNs for routing data messages for the user. For each HISN-HISN pair, one or more tunnels may exist. In our implementation, we utilize the PKI protocol [5] and a secret key to secure the tunnel. Figure 10 illustrates the message flow for the creation of a tunnel.

Tunneling procedure:

Step U1. The H_n sends a Create_Tunnel_Request(certificate, UserID, TunnelID) message to H_o. The message carries the certificate of the H_n, the user's ID, and a randomly generated tunnel ID.

Step U2. Upon receipt of Create_Tunnel_Request, H_o confirms the identity and validity of H_n by checking the certificate of H_n. If the certificate is valid, H_o generates

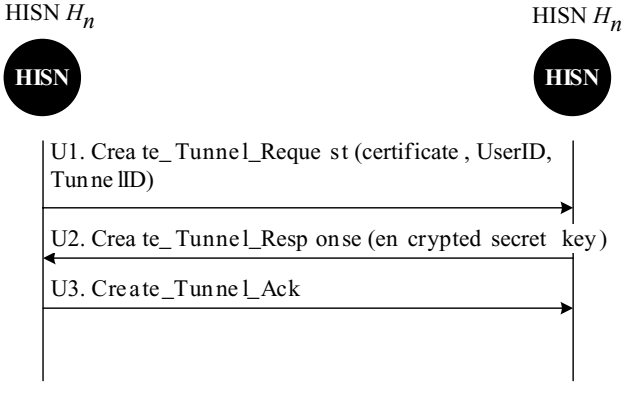


Fig. 10 The message flow for the tunneling procedure

a secret key to secure the tunnel. The secret key is transmitted as the user data, which is encrypted by the public key of H_n in the certificate. Then, H_o sends a Create_Tunnel_Response(encrypted secret key) message to H_n.

Step U3. Upon receipt of Create_Tunnel_Response message, H_n decrypts the data by using its own private key, and obtains the secret key. Then H_n sends a Create_Tunnel_Ack message to H_o for acknowledgment. Both H_n and H_o fill the tunnel's ID and the secret key information into the TunnelID and SecretKey fields of the Session contexts for the user, respectively. The Mobility Manager of H_o updates the CurrentHISN field in the user's UserProfile context as the IP address of H_n. Then H_n initiates a connection to H_o. At this moment, the tunnel has been established.

4.6. Application termination procedure

The IB executes the Application Termination procedure to tear down the activated sessions. Consider that a user has activated a session, S1, on H_o. During the session, the user roams to H_n, and then tear down the S_1 session at H_o. Figure 11 illustrates the message flow for the Application Termination procedure with details given below.

Step T1.1. The IB sends a Terminate_Application_Request (SessionID: S1) message to H_n.

Steps T2 and T3. The Mobility Manager of H_n gets the related information for the S1 session from the Visitor User Database by invoking the `getSessionInfo(SessionID: S1)` function.

Step T4. The Mobility Manager of H_n checks the LocatedHISN field in the Session context of S1 to obtain the IP address of the HISN where the S1 session is located. Then the Mobility Manager of H_n sends a Terminate_Application_Request(UserID, SessionID: S1) message to the Mobility Manager of H_o.

Steps T5 and T6. Upon receipt of the request, the Internet Service Module of H_o terminates the S_1 session. After successful termination of the application session, the Mobility Manager of H_o deletes the corresponding Session context from the Visitor User Database by invoking the `deleteSessionContext(S1)` function. The Visitor User Database returns a `success` value to the Mobility Manager.

Step T7. The Mobility Manager of H_o responds a Terminate_Application_Response(Session ID: S1) message to the Mobility Manager of H_n.

Steps T8 and T9. The Mobility Manager of H_n deletes the corresponding Session context for S1 from the Visitor User Database by invoking the `deleteSessionContext(S1)` function. The Visitor User Database returns a `success` value to the Mobility Manager of H_n.

Step T10. The H_n returns a Terminate_Application_Response Session ID: S1) message to the IB.

4.7. Logout procedure

The Logout procedure is executed to stop services in the HISN network. A user may activate different sessions on different HISNs. Before Logout, these sessions should be terminated. Without loss of generality, we assume that the user currently resides in H_n and has activated sessions on HISN H_n and HISN H_o. Figure 12 illustrates the message flow for the Logout procedure with details described in the following.

Step O1. The IB sends a Logout_Request message to H_n.

Steps O2 and O3. The Mobility Manager of H_n invokes the `getHISNList(UserID)` function to find the HISNs (that run the sessions for the user) by checking the LocatedHISN field in the Session contexts of the user. The Visitor

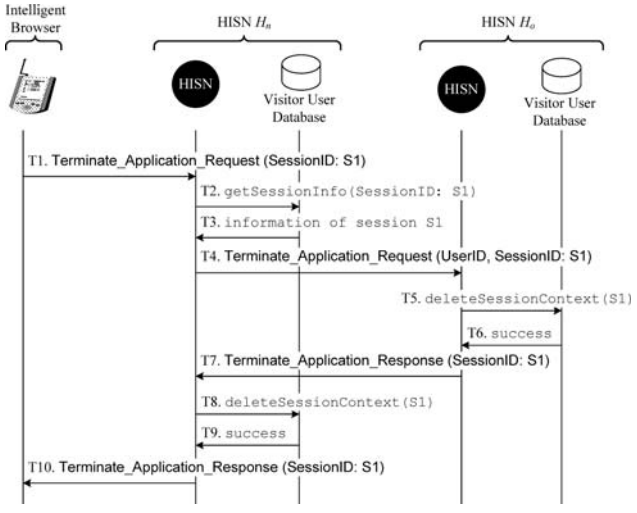


Fig. 11 The message flow for the Application Termination procedure

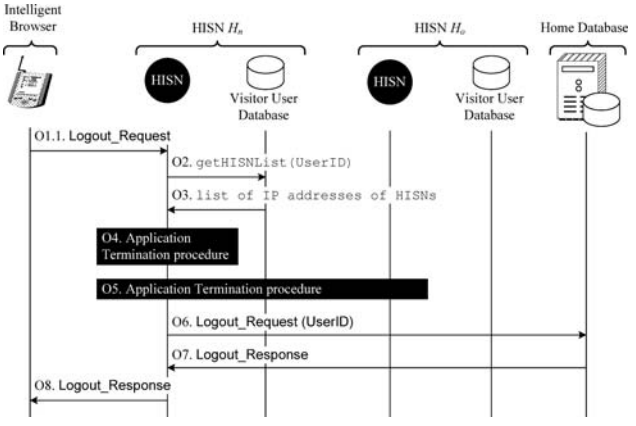


Fig. 12 The message flow for the Logout procedure

User Database returns the list containing the IP addresses of the HISNs.

Step O4. The Mobility Manager of H_n executes the Application Termination procedure to terminate the user's activated sessions in H_n.

Step O5. The Mobility Manager of H_n executes the Application Termination procedure to terminate the user's activated sessions in H_o.

Steps O6 and O7. The Mobility Manager of H_n and Home Database exchange the Logout_Request(UserID) and Logout_Response message pair to set the ServingHISN field of the user in the Home Database to be blank.

Step O8. A Logout_Response message is sent to the IB to indicate the result of the Logout procedure.

5. Performance evaluation

As shown in Fig. 9, when the user changes the connecting HISN node (from H_o to H_n), the Suspend procedure is executed to save and buffer the user data on the old HISN node H_o so that the data for the session is not lost, and the session can be continued on the new HISN node H_n. For the roaming users, extra system resource (e.g., memory) of H_o is consumed. To efficiently utilize the resource, we use the timer T_s to control the time when H_o should drop the related information for the roaming users. Obviously, if T_s is set smaller, the resource can be utilized more efficiently, however, this may cause more frequent session drops. Note that in the HISN network, we are concerned about the resource consumption issue more seriously due to that the distributed HISN network follows a non-continuing architecture, and the user may spend long time to roam among different HISNs or mobile devices. The resource of reservation on H_o may last for too long, which may cause the heavy penalty to the HISNs. This section studies how to properly set up the timer T_s so that we may balance the resource reservation penalty and the session dropping probability.

Suppose that during a running session, the mobile user roams N_r times. As shown in Fig. 9, at the ith roaming instant, let $t_{h,i}$ denote the period between the time when the Suspend procedure is completed and the time when the Login procedure is executed, $t_{l,i}$ be the total time required to finish the Login procedure, and $t_{r,i}$ be the total time required to finish the Resume procedure. Assume that t_s is the value of the timer T_s. In the Internet application server, a timer T_a starts when it receives the completion of the user packet. The application server expects to receive the next user packet before the T_a timer expires. If T_a expires, the server will quit the application. Let t_a be the value of the timer T_a. The roaming is successfully executed when the following condition holds:

$$t_{h,i} + t_{l,i} \leq t_s \text{ and } t_{h,i} + t_{l,i} + t_{r,i} \leq t_a \tag{1}$$

Otherwise, the roaming fails when one of the following two situations occurs:

$$t_{h,i} + t_{l,i} > t_s \quad \text{or} \quad t_{h,i} + t_{l,i} + t_{r,i} > t_a \tag{2}$$

For a session, we define a cost function for resource consumption as follows:

$$C_r = \sum_{i=1}^{N_r} C_{r,i} \tag{3}$$

where $C_{r,i}$ is the resource reservation penalty induced by the ith roaming. $C_{r,i}$ can be characterized by the following equation:

$$C_{r,i} = \begin{cases} t_{h,i} + t_{l,i}, & \text{if roaming successes} \\ \beta \times t_s, & \text{if roaming fails} \end{cases} \tag{4}$$

where β is a penalty factor for an unsuccessful roaming. The rationale behind (4) is as follows. If the roaming is successfully performed, H_o will reserve resource for the roaming session for $t_{h,i} + t_{l,i}$. If the roaming fails, H_o will reserve resource for the roaming session for t_s. In this case, we time t_s by β as the cost for the roaming session.

We adopt the event-driven based simulation technique, which is similar to that used in [3], and the details are not presented here. Suppose that the $t_{h,i}$, $t_{l,i}$, and $t_{r,i}$ are exponentially distributed with means $\frac{1}{\mu_h}$, $\frac{1}{\mu_l}$, and $\frac{1}{\mu_r}$, respectively. We assume that the residence time for the service area of an HISN are Gamma distributed with mean $\frac{1}{\eta_a}$ and variance $v_a = \frac{1}{\eta_a^2 \alpha}$, where α is the shape parameter. The Gamma distribution is selected because it can be shaped to represent many distributions [11]. In our study, we run simulation experiments to measure two performance indicators: the resource reservation cost C_r for a session and the session dropping probability P_d which is defined as the probability that a session cannot be completed due to the unsuccessful roaming. We simulate 1,000,000 sessions in an experiment to ensure the stability of the simulation results. We investigate the effects of the T_s timer setup on the P_d and C_r performance for two kinds of Internet application sessions: FTP and Telnet.

In this study, we set $\frac{1}{\mu_h} = 50$ seconds, $\frac{1}{\mu_l} = 1$ second, $\frac{1}{\mu_r} = 1$ second, $\frac{1}{\eta_a} = 900$ seconds, and $\alpha = 4$. For other parameter setup, we observe the similar phenomena.

Effects of T_s Setup for FTP Sessions: The FTP application is designed for file transfer, which bears long transmission time for data. In our study, we assume that the service times for the FTP applications are exponentially distributed with mean $\frac{1}{\mu_F}$. Based on the measured data in [19], we set $\frac{1}{\mu_F} = 429$ seconds. In most application programs of FTP, the length of the T_a timer for the FTP

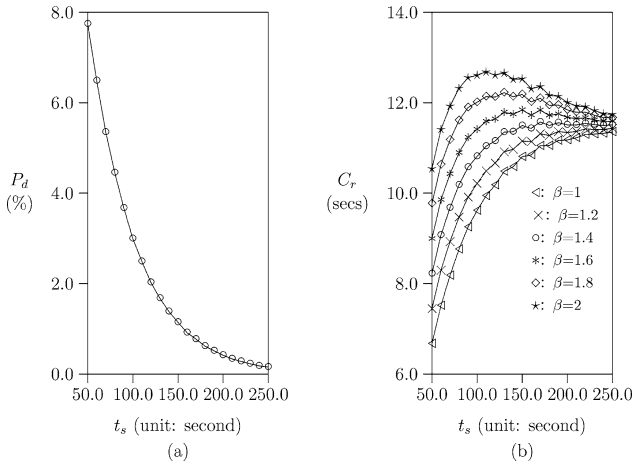


Fig. 13 Effects of T_s setup for FTP sessions ($\frac{1}{\mu_h} = 50$ secs; $\frac{1}{\mu_l} = 1$ sec; $\frac{1}{\mu_r} = 1$ sec; $\frac{1}{\eta_a} = 900$ secs; $\alpha = 4$; $\frac{1}{\mu_F} = 429$ secs; $t_a = 1800$ secs)

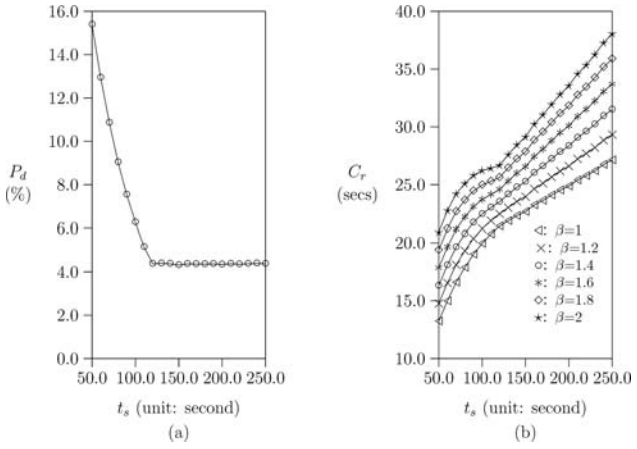


Fig. 14 Effects of T_s setup for telnet sessions ($\frac{1}{\mu_h} = 50$ secs; $\frac{1}{\mu_l} = 1$ sec; $\frac{1}{\mu_r} = 1$ sec; $\frac{1}{\eta_a} = 900$ secs; $\alpha = 4$; $\frac{1}{\mu_T} = 700$ secs; $t_a = 120$ secs)

application is $t_a = 1800$ seconds, which is adopted in this study. Figure 13 plots P_d and C_r as functions of t_s for the FTP session.

Figure 13 (a) shows that as t_s increases, P_d significantly decreases. This phenomenon reflects the fact that as the timer T_s is longer, the session has better chance to be completed. When $t_s > 80$ seconds, P_d is down below 5%, which satisfies the QoS requirement.

In Fig. 13 (b), we observe that as $\beta \leq 1.4$ (i.e., low penalty), the C_r cost increases as t_s increases. On the other hand, when $\beta > 1.4$ (i.e., high penalty), as t_s increases. the C_r cost significantly increases, and then slightly decreases. The disjunctive point occurs at $t_s = 120$ seconds. When t_s is small (i.e., $t_s \leq 120$ seconds in this figure), P_d is larger, it is more likely that roaming fails. In this situation, from (4), the $C_{r,i}$ cost is dominated by the β penalty factor. On the other hand, a larger t_s causes smaller P_d values (i.e., $t_s > 120$ seconds in this figure). The roaming has better chance to be successfully completed, and from (4), the $C_{r,i}$ cost is dominated by $t_{h,i} + t_{l,i}$. Thus, we observe the above phenomenon.

Based on the two phenomena in Fig. 13, for the FTP application, when $\beta \leq 1.4$, we may set $t_s = 80$ seconds. Since P_d is below 5% as $t_s \geq 80$ seconds, and C_r increases as t_s increases, to satisfy the QoS requirement and to lower the C_r network cost, $t_s = 80$ seconds is the best choice. When $\beta > 1.4$, we set t_s as large as possible to minimize both P_d and C_r.

Effects of T_s Setup for Telnet Sessions: Telnet is an interactive application. The communication between the user and the application server has strict delay requirement. As the T_a timer is set smaller, the application server has more chance to quit the application due to no response. With smaller T_a setup, we may more strictly bound the delay for the communication between the user and the application server, and it is clearer to observe the performance of the HISN network for the Telnet application. In this study, we set $t_a = 120$ seconds (typically, in most programs, the length of the T_a timer for the Telnet application server is $t_a = 600$ seconds). Suppose that the service times for the Telnet applications are exponentially distributed with mean $\frac{1}{\mu_T}$. We set $\frac{1}{\mu_T} = 700$ seconds, which is the same as the measured data in [19]. Figure 14 examines the effects of the timer T_s setup on P_d and C_r for the Telnet application.

In Fig. 14 (a), we observe that when $t_s \leq t_a$ (i.e., $t_s \leq 120$ seconds in this figure), the dropping probability P_d significantly decreases as t_s increases. When $t_s > t_a$ (i.e., $t_s > 120$ seconds), the P_d values are almost the same for the different t_s setups and are less than 5%. From (1), we know that the roaming is successfully executed, when the following condition holds.

$$t_{h,i} + t_{l,i} \leq t_s \quad \text{and} \quad t_{h,i} + t_{l,i} \leq t_a - t_{r,i}$$

By considering the above condition, when $t_s \leq t_a - t_{r,i}$, the probability for successful roaming can be $\Pr[t_{h,i} + t_{l,i} \leq t_s]$. This implies that the P_d value is only affected by the t_s setup, which reflects what we observe in Fig. 14 (a) when $t_s \leq 120$ seconds (i.e., $t_s \leq t_a$). On the other hand, when $t_s > t_a$, the probability for successful roaming is $\Pr[t_{h,i} + t_{r,i} \leq t_a - t_{r,i}]$, which implies that the P_d value is only affected by the t_a setup. In our study, t_a is set as a fixed value 120 seconds. Thus, in Fig. 14 (a), when $t_s > 120$ seconds, we observe that the P_d values are the same for different t_s setups.

As shown in Fig. 14 (b), as t_s increases, the C_r cost increases. With various β setups (from 1 to 2), we observe the same phenomena. To conclude, P_d is below 5% as $t_s \geq 120$ seconds, and C_r increases as t_s increases. To

satisfy the QoS requirement and to lower the C_r network cost, we may select $t_s = 120$ seconds.

6. Conclusion

In this paper, we first examine the important issues for the integration of heterogeneous wireless networks in B3G. Then we design and implement a Heterogeneous network Support Node (HISN) network platform that accommodates the required functionalities for such an integration: protocol conversion, session mobility, personal mobility, and terminal mobility. Our design is transparent to the bearer networks, which may reduce the deployment cost for the HISN platform. Our design focuses on the application-level integration in the sense that all HISN nodes form the edge access points bridging end users to the Internet, which is consistent with the design philosophy of wireless mesh networks.

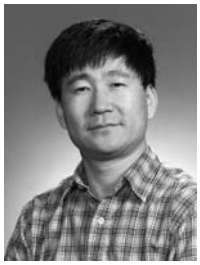
Acknowledgments The authors would like to express their sincere appreciation to the anonymous reviewers whose comments have significantly improved the quality of this paper.

References

1. World Wide Web Consortium (W3C). Extensible Markup Language (XML) 1.1. Technical Report (Feb. 2004).
2. 3GPP. Virtual Home Environment (VHE)/Open Service Access (OSA); Stage 2. Technical Report Technical Specification 3G TS 23.127 version 6.0.0 (2003-01) (2003).
3. P. Lin, Y.-B. Lin and I. Chlamtac, "Modeling frame synchronization for UMTS high-speed downlink packet access," *IEEE Transactions on Vehicular Technology* 52(1) (Jan. 2003) 132–134.
4. B. Thai, R. Wan, A. Seneviratne and T. Rakotoarivelo, "Integrated personal mobility architecture: a complete personal mobility solution," *ACM Mobile Networks and Applications* 8(1) (Feb. 2003) 27–36.
5. R. Housley, W. Polk, W. Ford and D. Solo, "Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," Technical Report RFC 3280, Internet Engineering Task Force (April 2002).
6. C.E. Perkins, IP Mobility Support for IPv4. "Technical report RFC 3344," Internet Engineering Task Force (Aug. 2002).
7. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "SIP: session initiation protocol. technical report RFC 3261," Internet Engineering Task Force (June 2002).
8. World Wide Web Consortium (W3C). "Extensible hyperText markup language (xHTML) version 1.0." Technical Report (Aug. 2002).
9. 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; General Packet Radio Service (GPRS); Service Description; Stage 2. Technical Report Technical Specification 3G TS 23.060 version 4.1.0 (2001-06) (2001).
10. World Wide Web Consortium (W3C). "Synchronized multimedia integration language (SMIL 2.0)," Technical Report (Aug. 2001).
11. A.-M. Law and W.-D. Kelton, "Simulation modeling and analysis," McGraw-Hill, 3rd Edition (2000).
12. B. Raman, R.H. Katz and A.D. Jose, "Universal inbox: providing extensible personal mobility and service mobility in an integrated communication network," *Proceedings of Workshop Mobile Computing Systems and Applications (WMSCA'00)* (Aug. 2000).
13. H.J. Wang, B. Raman, C.-N. Chuah, R. Biswas, R. Gummadi, H. Barbara, X. Hong, E. Kiciman, Z. Mao, J.S. Shih, L. Subramanian, B.Y. Zhao, A.D. Joseph and R.H. Katz, "ICEBERG: an internet-core network architecture for integrated communication," *IEEE Personal Communications* 7(4) (Aug. 2000) 10–19.
14. P. Maniatis, M. Roussopoulos, E. Swierk, K. Lai, G. Appenzeller, X. Zhao and M. Baker, "The mobile people architecture," *ACM Mobile Computing and Communications Review* 3(3) (July 1999) 36–42.
15. M. Roussopoulos, P. Maniatis and E. Swierk, "Person-level routing in the mobile people architecture," *Proceedings of the USENIX Symposium on Internet Technologies and Systems* (Oct. 1999).
16. Wireless WAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: Higher Speed Physical Layer (PHY) Extension in the 2.4 GHz Band. Technical Report, IEEE (1999).
17. World Wide Web Consortium (W3C) . XSL Transformations (XSLT) Version 1.0. Technical Report (Nov. 1999).
18. ETSI SMG. User of Data Terminal Equipment-Data Circuit Terminating; Equipment (DTE-DCE) Interface for Short Message Service (SMS) and Cell Broadcast Service (CBS) (GSM 07.05 version 5.3.0). Technical Report Recommendation GSM 07.05, ETSI/TC (1997).
19. J. Crowcroft and I. Wakeman, "Traffic analysis of some UK-US academic network data" *Proceedings of INET'91* (June 1991).
20. J. Postel and J. Reynolds, "FILE TRANSFER PROTOCOL (FTP)," Technical Report RFC 959, Internet Engineering Task Force (Oct. 1985).
21. The Ninja Project. http://ninja.cs.berkeley.edu/.
22. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wcesdkr/html/_wcesdk_win32_getsysteminfo.asp.
23. http://www.bluetooth.com/.
24. http://www.irda.org/.
25. http://www.wapforum.org/.

Phone Lin (M'02-SM'06) received his BSCSIE degree and Ph.D. degree from National Chiao Tung University, Taiwan, R.O.C. in 1996 and 2001, respectively. From August 2001 to July 2004, he was an Assistant Professor in Department of Computer Science and Information Engineering (CSIE), National Taiwan University, R.O.C. Since August 2004, he has been an Associate Professor in Department of CSIE and Graduate Institute of Networking and Multimedia, National Taiwan University, R.O.C. His current research interests include personal communications services, wireless Internet, and performance modeling. Dr. Lin is an Associate Editor for IEEE Transactions on Vehicular Technology, a Guest Editor for IEEE Wireless Communications special issue on Mobility and Resource Management, and a Guest Editor for ACM/Springer MONET special issue on Wireless Broad Access. He is also an Associate Editorial Member for the WCMC Journal. P. Lin's email and website addresses are plin@csie.ntu.edu.tw and http://www.csie.ntu.edu.tw/∼plin, respectively.

Huan-Ming Chang received the BSCSIE degree and Master CSIE degree from National Taiwan University, R.O.C. in 2003 and 2005, respectively. His current research interest includes wireless Internet. H.-M. Chang's email address is r91114@csie.ntu.edu.tw.



Yuguang Fang received a Ph.D. degree in Systems and Control Engineering from Case Western Reserve University in January 1994, and a Ph.D. degree in Electrical Engineering from Boston University in May 1997. From June 1997 to July 1998, he was a Visiting Assistant Professor in Department of Electrical Engineering at the University of Texas at Dallas. From July 1998 to May 2000, he was an Assistant Professor in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology. In May 2000, he joined the Department of Electrical and Computer Engineering at University of Florida where he got the early promotion to Associate Professor with tenure in August 2003 and to Full Professor in August 2005. He has published over 180 papers in refereed professional journals and conferences. He received the National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002. He is currently serving as an Editor for many journals including IEEE Transactions on Communications, IEEE Transactions on Wireless Communications, IEEE Transactions on Mobile Computing, and ACM Wireless Networks. He is also actively participating in conference organization such as the Program Vice-Chair for IEEE INFOCOM'2005, Program Co-Chair for the Global Internet and Next Generation Networks Symposium in IEEE Globecom'2004 and the Program Vice Chair for 2000 IEEE Wireless Communications and Networking Conference (WCNC'2000).

Shin-Ming Cheng received the BSCSIE degree in 2000 from National Taiwan University, Taiwan, R.O.C., where he is currently working toward the Ph.D. degree in the Department of Computer Science and Information Engineering, National Taiwan University. His current research interests include mobile computing, personal communications services, and wireless Internet. S.-M. Cheng's email and website addresses are shimi@pcs.csie.ntu.edu.tw and http://www.pcs.csie.ntu.edu.tw/∼shimi, respectively.