Applied Mathematics & Information Sciences
An International Journal

Design and Implementation of Simulation Testing System for GPRS and UMTS 3G Core Network

Xia Hailun

Beijing Key Laboratory of Network System Architecture and Convergence, School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876 P. R. China

Received: 13 Sep. 2012, Revised: 20 Nov. 2012, Accepted: 1 Dec. 2012
Published online: 1 Feb. 2013

Abstract: In order to improve the efficiency of maintaining packet core network and test the network elements, a simulation testing system for GPRS and UMTS 3G packet core network is proposed. The system can simultaneously simulate several network elements to connect with SGSN, GGSN, WAP gateway respectively for testing. In the system, multiple UEs can be simulated, and multiple applications can run on each UE. Protocol Stack Pool is developed to accommodate several protocol stacks at Gb, Iu-PS, Gn, Gi interfaces. Simulated UEs can communicate with the network side equipments through the Protocol Stack Pool. The system was implemented in Linux operating system and a web-based management subsystem was provided. The system was tested by Tektronix K1297 testing equipment and by accessing the current cellular mobile communications network. We also emulate the main network elements and set up an emulated core network to test in laboratory. The testing results show that applications can run on the system to test the network elements.

Keywords: core network, simulation testing, simulation testing system

1. Introduction

In recent years, with the rapid development of packet service in cellular mobile communications network, especially the development of wireless application protocol(WAP) service and mobile Internet, the packet domain of the general packet radio service (GPRS) and the universal mobile telecommunications system (UMTS) 3G core network becomes more important and more complicated. The maintenance of the core network also becomes complicated. There are a large number of network elements in the packet core network. Each network element is so important that once error occurs, it could be a significant error, and many users could be affected. It will be lagged if fault is found by users complaints. To ensure the packet core network to work normally, huge human and material resources should be poured if only traditional terminal testing methods are adopted. And the testing functionality is limited. The operators are lack of method of fault warning and rapid fault locating.

There are some commercial testing instruments of some equipment manufacturers, such as Tektronix Company, Sprint Company and Agilent Company. But they are all expensive and hard to operate. In recent years, some open source projects appear, like Osmocom OpenBSC [1], OpenGGSN [2]. These projects implement the main functionality of the base station controller (BSC) and the gateway GPRS support node (GGSN) in the form of software. But they can only emulate just few network elements simply, and cant be used to test the network elements.

In this paper, to improve the efficiency of maintaining packet core network and test the network elements, a simulation testing system for GPRS and UMTS 3G packet core network, which is named CNSimTest system, is designed and implemented in Linux operating system. Going beyond the air interface, the CNSimTest system can simultaneously simulate the packet control unit (PCU) of GPRS network, the radio network controller (RNC) of UMTS 3G network, the serving GPRS support node (SGSN) and GGSN to access SGSN, GGSN, WAP gateway for testing through Gb or Iu-PS interface, Gn interface and Gi interface respectively. The system supports simulating multiple user equipment (UE) and simulating multiple applications on each UE. Thus, we can also test the performance of some packet service. The

* Corresponding author e-mail: xiahailun@gmail.com

system provides an economy way to test the packet core network and locate the faulty network element. Also, the system can be used to setup a core network testbed for study and research.

This paper is organized as follows. In Section 2, the proposed CNSimTest system is presented, including the system application environment, the system architecture and the protocol stack of the system. The implementation details are described in Section 3. In Section 4, testing results in emulated core network and current cellular mobile communications network are provided. Finally, conclusions are drawn in Section 5.

2. System Design

2.1. System application environment

In GPRS and UMTS 3G packet core network, there are some primary network elements such as SGSN, GGSN, Radius Server, WAP gateway, etc. In GPRS network, PCU and SGSN are connected through Gb interface. In UMTS 3G network, RNC and SGSN is connected through Iu-PS interface. In order to simultaneously support testing the SGSNs in GPRS and UMTS 3G, CNSimTest implements both Gb interface and Iu-PS interface. Because Frame Relay equipments are need at Gb over Frame Relay interface, and Gb over IP is more and more widely deployed, our system only focus on Gb over IP interface. So in order to test the network elements of packet core, our CNSimTest system should simulate the protocols of PCU, RNC, SGSN, GGSN, access the core network through multiple interfaces (i.e. Gb over IP interface, Iu-PS interface, Gn interface and Gi interface), and selectively connect to certain network element to launch the testing tasks, as shown in Figure 1.

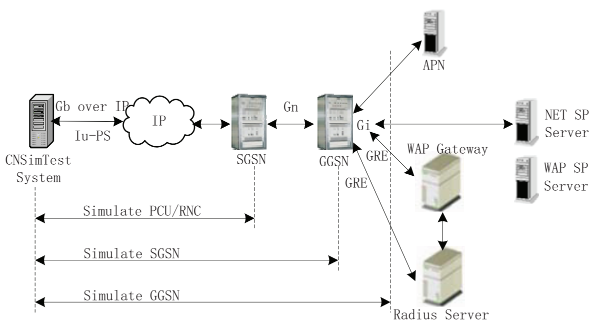

Figure 1 The application environment of CNSimTest system

CNSimTest can also be used to locate the faulty network element. For example, if error occurs in some packet radio service in UE, we can use CNSimTest system to test SGSN. If no error occurs, we can tell that there must be something wrong with the Radio Access Network (RAN). If error occurs, we can continue testing GGSN through Gn interface. Some problem should occur in SGSN if no error occurs. Thus, we can rapidly locate the faulty network element when there is something wrong with the packet core network.

To launch the testing task, UE and application should be simulated in CNSimTest system. Multiple UEs and multiple applications are simulated to test the network elements. By analyzing the testing results, the system can also evaluate the performance of the network elements and the applications.

2.2. CNSimTest System Architecture

The architecture of the CNSimTest system is shown in Figure 2. The system consists of three main parts, web-based management subsystem, UE Pool and Protocol Stack Pool. The web-based management subsystem control the UEs in UE Pool to run testing tasks, and the application data of UE should be transferred to the tested network element through the corresponding protocol stack in Protocol Stack Pool.

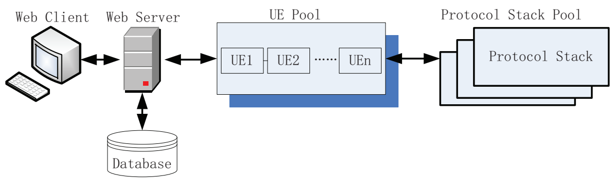

Figure 2 The architecture of the CNSimTest system

- *web-based management subsystem*. The system includes web client, web server and database. It is responsible for configuration management, controling of testing tasks and fault warning. The database stores the UE information, configuration parameters, data of testing task, etc.
- *UE Pool*. Multiple UEs can be simulated in the UE Pool. Each UE can run multiple applications. UE should have its own IMSI, MSISDN, PDP address, as actually it is.
- *Protocol Stack Pool*. There are multiple protocol stacks in the Protocol Stack Pool. Because the CNSimTest system needs to access SGSN, GGSN, WAP gateway through Gb or Iu-PS, Gn, Gi interface respectively, the corresponding protocol stack should be in the pool. Protocol Stack Pool receives the application data from the UE Pool and then sends to the tested networks element through its member protocol stack.

The UE Pool and Protocol Stack Pool can communicate by TCP/IP. Thus, UE Pool and Protocol Stack Pool can resident in one equipment or two independent equipments.

2.3. Protocol stack design

As for the actual UE, the application data encapsulated by TCP/IP is transferred through the lower protocol of air interface. For example, in GRPS network, the protocol stack of the actual UE is shown as Figure 3. the IP-encapsulated application data is sent to the Sub Network Dependent Convergence Protocol (SNDCP) protocol, and then to the Logic Link Control (LLC) protocol.

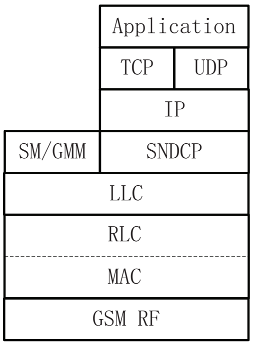

Figure 3 The protocol stack of actual UE

In our CNSimTest system, air interface is stepped over. Two main aspects of the system are considered: application and the lower protocol stack. The simulated UEs only focus on application, and the lower protocol stack is put at Protocol Stack Pool. User Datagram Protocol (UDP) is used to transfer the encapsulated application data between UE Pool and Protocol Stack Pool. By this way, UE Pool and Protocol Stack Pool can be separate and easily deployed. The protocol stack of a single simulated UE in UE Pool is designed as shown in Figure 4. The Session Management (SM), GPRS Mobility Management (GMM) [3][4][5], SNDCP and LLC protocols are all put at Protocol Stack Pool. Thus, the simulated UE is independent with the lower protocol stack. That is to say, the simulated UE can be used by both GPRS and UMTS 3G protocol stack.

The Protocol Stack Pool should support Gb over IP, Iu-PS, Gn and Gi interfaces. So there are several protocol stacks coexisting in the Protocol Stack Pool, as described in Figure 5. We move the SM/GMM protocol of UMTS, which exists in actually UE, on the protocol stack of Iu-PS interface. And we also move SM/GMM, SNDCP and LLC of GPRS network, on the protocol stack of Gb interface. Thus our CNSimTest system can access the

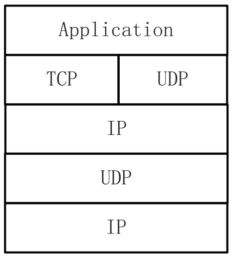

Figure 4 The protocol stack of single simulated UE

network side elements. In the data plane, GTP-U[6][7] is shared between Iu-PS[8] interface and Gn interface, because the port used by GTP-U is fixed.

The SM/GMM, SNDCP and LLC protocols are moved from UE protocol. These protocols originally only support the Finite State Machine (FSM) of a single UE. In our CNSimTest system, we make some revisions to support multiple UEs.

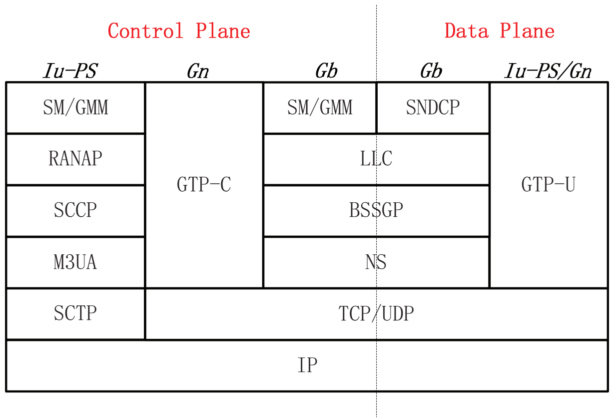

Figure 5 The protocol structure of Protocol Stack Pool

3. System Implementation Details

The CNSimTest system is developed in Linux operating system. C programming language is used. The system supports multiple applications simulation in UE, multiple UE simulation, multiple RNCs simulation, and multiple PCUs simulation. PCU, RNC, SGSN and GGSN can be simulated at the same time in CNSimTest system, to access SGSN, GGSN, WAP gateway over IP for testing if needed. Details of implementation are described in the following.

3.1. Web-based management subsystem

In the web-based management subsystem, which is developed in J2EE environment, besides configuration management, a control center is developed in C programming language behind the web server to control the testing tasks, such as initiating UE and its applications. The IMSI, MSISDN, type and configuration parameters of UEs are stored in the database, which can be sent through the control center to the UEs in UE pool, and can be used by UE to register in Protocol Stack Pool before testing task begins.

3.2. UE pool

In UE Pool, multiple UEs can be simulated. And many applications can run in a single UE. The application data should be encapsulated by TCP/IP, and then be transferred to the Protocol Stack Pool using UDP socket. The structure of a single UE is showed in Figure 6. We enhance the LWIP[9] protocol stack to encapsulate the application data. LWIP is a lightweight TCP/IP protocol stack used in many embedded system. We add a new netif driver named socketif, which runs in a separate thread, in order to send the LWIP-encapsulated IP packets using TCP/IP of Linux operating system, and forward the data received from UDP socket of Linux to LWIP. Thus, we can achieve TCP/IP over UDP. The socket interface of LWIP is similar to BSD socket, so the application service can be easily developed as using BSD socket. We need not change the code of sending and receiving function, and only need to change the include files of code files.

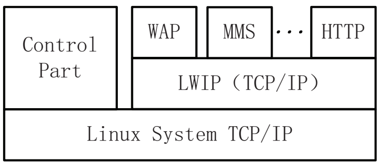

Figure 6 Structure of a single simulated UE

Control part of a UE is responsible for controlling UE and its applications, such as UE registering and sending control signaling messages to the Protocol Stack Pool, for examples, UE attach and activation messages. After activation, each UE can get an IP address from network, which is used to initiate the LWIP protocol stack. Thus, the application data of different UE can be packaged by different source IP address. The control part is also responsible for controlling the authentication of UE. A3 algorithm of SIM card and f1, f2, f3, f4, f5 algorithms of USIM card are implemented in the control part, which can communicate with GMM protocol in the Protocol Stack Pool for UE authentication. Thus, virtual SIM and USIM can be achieved.

The applications are developed in C programming language such as WAP, MMS (Multimedia Messaging Service), HTTP, FTP, PING, etc. Each UE is implemented as a process in Linux, and each application, control part, LWIP protocol stack run as different threads in the process. Thus, the UEs in UE pool run as different processes in Linux. These processes can send and receive data using different UDP socket (i.e. UDP port) of Linux.

3.3. Protocol Stack Pool implementation

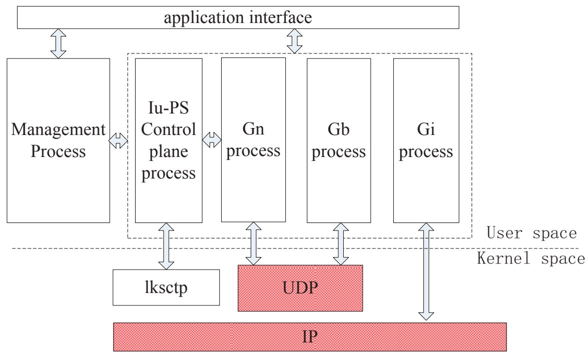

Figure 7 Implementation of Protocol Stack Pool

We develop the Protocol Stack Pool in C programming language in user space of Linux operating system, as shown in Figure 7. Application interface send and receive data and signaling message between UE Pool and Protocol Stack Pool. Management process is developed to manage the protocols and processes, warn if error occurs. Iu-PS control plane, Gn, Gb and Gi are implemented in separate processes. The protocols at one interface are implemented in one process. GTP-U is included in Gn process and shared by Iu-PS interface as data plane.

3.3.1. Application interface

At application interface, we use UDP socket between UE pool and Protocol Stack Pool to transfer application data. Each UE has its own listening ports for data and signaling message, which are carried to Protocol Stack Pool when UE register in certain process in Protocol Stack Pool. Thus, Protocol Stack Pool can communicate with the corresponding UE. We also define different types of signaling message for UE to trigger the signaling procedure.

3.3.2. GPRS Gb interface implementation

We only implement the Gb over IP interface, and have developed SM/GMM, SNDCP, LLC, BSSGP and NS protocols. All the protocols run in one process of Linux. Each protocol layer provides one interface function for its upper layer and the other for its lower layer. NS control sublayer uses UDP socket to send packets.

Actually, each UE has its own SM/GMM. To support multiple UEs, we maintain an array of SM/GMM structures in the SM/GMM layer. Each structure is used to maintain the corresponding UEs states. SM/GMM is controlled by UE in UE pool. GMM can communicate with the UE to initiate the authentication procedure.

To support multiple PCUs simulation for testing multiple SGSNs, we design the PCU registration procedure between UE and Gb interface. SM/GMM, BSSGP and NS maintain an array of PCU structures respectively.

3.3.3. Iu-PS interface implementation

The control plane of Iu-PS is implemented on Sigtran[10]. We adopt lksctp[11] as SCTP (Transmission Protocol)[12] protocol in transport layer, which is already included in Linux kernel 2.6 and the later versions. SM/GMM, Radio Access Network Application Part (RANAP)[13], Signaling Connection Control Part (SCCP) and M3UA (MTP3 User Adaptation)[14] are implemented in one process of Linux, like Gb interface. SM/GMM is implemented as the same way as in Gb interface.

In order to test multiple SGSNs at the same time, we should simulate multiple RNCs, connecting to SGSNs one to one for testing. Each RNC has its own SCTP association and M3UA application. We maintain multiple MTP structures in M3UA layer and multiple Signaling Point structures in SCCP layer, which can correspond to SCTP association at last. As for RANAP, we store the information of each RNC (such as RNC ID, RNC IP address, etc.) in a structure, and maintained multiple structures for the RNCs.

GTP-U is used as the data plane of Iu-PS interface. Because the port used by GTP-U is fixed in GTP protocol, and both Iu-PS and Gn use the GTP-U, we share the GTP-U module between Iu-PS and Gn, as described in the following Gn interface implementation.

3.3.4. Gn interface implementation

CNSimTest system uses GTP to test GGSN, including GTP-C, GTP-U. We enhance the GTP library of OpenGGSN, which is an open source project. As for GTP-C, we also support version 0 and version 1. The signaling procedures such as create PDP context request , delete PDP context request, are triggered by the message from the control part of UE in UE pool.

GTP-U module is shared between Iu-PS interface and Gn interface. In order to test several GSN (including SGSN and GGSN) at the same time, we store the remote GSN address in PDP context, which is carried during UE registration procedure of Iu-PS interface and Gn interface. We adopt two hash tables to manage the PDP contexts of Iu-PS and Gn respectively. The two hash tables are maintained by SM and GTP-C respectively. The local TEIDs of the PDP context allocated by RANAP and GTP-C are assigned within different range. When receiving a packet, GTP-U gets TEID from the packet, and selects a hash table according to the TEID range. By seeking the hash table, GTP-U can get the listening port of the destination UE, and then forward the packet to the destination UE. When sending a packet, GTP-U lookup the TEID and the remote GSN address in the corresponding hash table according to the UE IP address (i.e. source IP address of the packet), encapsulate the GTP-U packet header, and send the packet to the corresponding GSN.

3.3.5. Gi interface implementation

Through Gi interface, CNSimTest system connects to the WAP gateway. CNSimTest system should communicate with Radius server for billing by using Radius protocol. We configure GRE tunnel to connect to WAP gateway and Radius server. It also should emulate the function of GGSN, such as IP address allocating and data forwarding.

Referring to the OpenGGSN project, we developed our Gi interface. We adopt two IP address pools based on hash table to manage the UEs IP addresses, one for WAP service, and the other for NET service. When UE in UE pool registers to the Gi module, an IP address is allocated and added to hash table.

Data from different UEs (i.e. from different source IP address to WAP gateway) should be forwarded by Gi module, and data from WAP gateway (i.e. to different destination IP address of UE) should be forwarded to the corresponding UE, so we must emulate a virtual router in Gi module. TUN virtual NIC is used to emulate a virtual router for data forwarding, like OpenGGSN. The ability of forwarding data of Linux operating system must be enabled, otherwise the system would not forward data like a router. When application data from UE comes down, we directly send the data to the TCP/IP protocol stack of Linux system, and then the data would be routed to the WAP gateway. When data from WAP gateway is received, it would be routed to the TUN. We can receive the data from TUN, and lookup the destination IP address in the hash table to get the UEs information. Then the data can be sent to the corresponding UE.

4. System Testing

In this section, to evaluate the function of CNSimTest system, we use K1297-G35 testing equipment of Tektronix Company, and current cellular mobile communications network.

4.1. Testing using K1297-G35 instrument

K1297-G35 is a powerful protocol testing instrument of Tektronix Company. In the phase of developing, it is used to test our protocols in Protocol Stack Pool. As an example, testing of attach procedure at Iu-PS interface is shown in Figure 8.

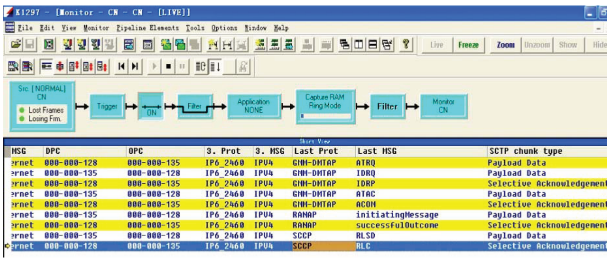

Figure 8 Attach procedure testing

Here, G35 is used as an emulated SGSN. We can see that the attach procedure between CNSimTest system and K1297-G35 is ATRQ (Attach Request)→ IDRQ (Identity Request) → IDRP (Identity Response)→ ATAC(Attach Accept) → ACOM(Attach Complete). It conforms with the original protocol.

4.2. Testing using EmuCN

We set up an emulated core network, named EmuCN, including SGSN emulator, GGSN emulator, Radius server and application server. We adopt OpenGGSN as the GGSN emulator, part of OsmoCom OpenBSC as the GPRS SGSN emulator. We also develop the 3G SGSN emulator at Iu-PS interface. The EmuCN is described in Figure 9.

As an example, Figure 10 shows the attach procedure testing in EmuCN. The procedure is between our CNSimTest system and 3G SGSN emulator. The IP address and signaling point code of CNSimTest system are 192.168.11.101 and 135 respectively. The IP address and signaling point code of 3G SGSN emulator are configured to 192.168.11.1 and 128. We can see that the attach procedure is complete, and the sccp connection is released at last.

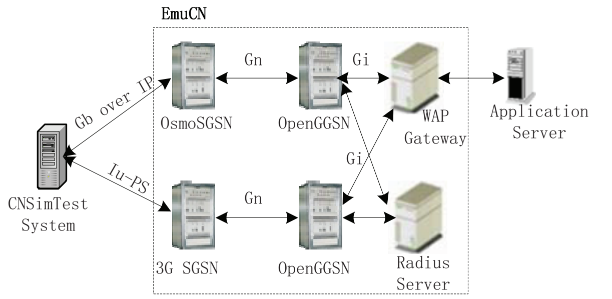

Figure 9 Structure of EmuCN

Source	Destination	Protocol	Info
135	128	RANAP	id-InitialUE-Message (DTAP) (GMM) Attach Request
192.168.11.1	192.168.11.101	SCTP	SACK
128	135	SCCP (Int. ITU)	CC
128	135	RANAP	id-CommonID
128	135	RANAP	id-DirectTransfer (DTAP) (GMM) Attach Accept
192.168.11.101	192.168.11.1	SCTP	SACK
135	128	RANAP	id-DirectTransfer (DTAP) (GMM) Attach Complete
192.168.11.1	192.168.11.101	SCTP	SACK
128	135	RANAP	id-Iu-Release
135	128	RANAP	SACK id-Iu-Release
192.168.11.1	192.168.11.101	SCTP	SACK
128	135	SCCP (Int. ITU)	RLSD
135	128	SCCP (Int. ITU)	SACK RLC

Figure 10 Attach procedure testing in EmuCN

4.3. Testing using the current cellular mobile communications network

We also access the current GPRS and TD-SCDMA core network of China Mobile Limited Company to test our system. The TD-SCDMA core network is as the same as the UMTS 3G core network. We have test the SGSN, GGSN, WAP gateway of the current cellular network through Gb, Iu-PS, Gn, Gi interfaces. All protocol procedures are verified correctly and applications can also run normally.

As an example, Figure 11 shows the PDP context activation procedure at Gn interface. The packets are grabbed in wireshark software, which is a widely-used protocol analyzer. From Fig.9, we can see that CNSimTest system sent Create PDP context request message to GGSN (IP address is 221.177.0.241) in the existing network. GGSN responded a message with a PDP address 10.3.17.110, which is allocated to a UE. Then the UE use this IP address to initiate HTTP application. The application data was carried over GTP-U.

Figure 12 shows the testing procedure at Gi interface. We can see that CNSimTest system sent Accounting Request message to Radius server. The IP address 172.16.2.7 of UE , which is allocated by CNSimTest,is carried in the message. The Radius server responded with accounting start message. Then the UE started an HTTP application with the WAP gateway (IP address is 10.0.0.172).

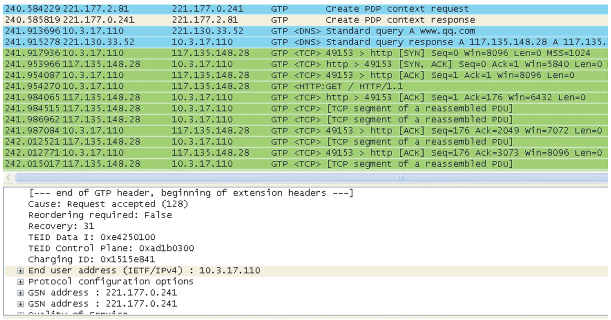

Figure 11 Gn interface testing

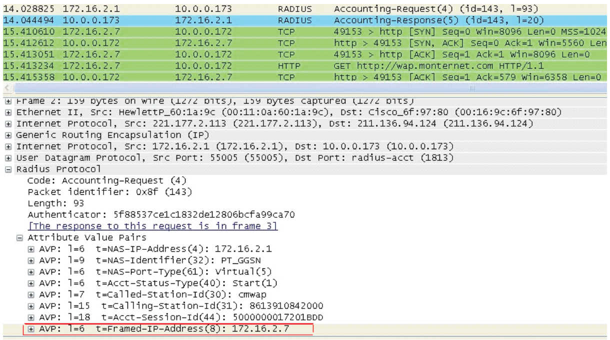

Figure 12 Gi interface testing

5. Conclusion

In this paper, we propose a simulation testing system for GPRS and WCDMA packet core network. The system can simulate PCU, RNC, SGSN, GGSN to connect with SGSN, GGSN, WAP gateway respectively for testing. UE Pool and Protocol Stack Pool are designed to communicate with network elements. In UE Pool, multiple UEs can be simulated, and multiple applications can run on each UE. The protocol stacks of Gb, Iu-PS, Gn, Gi interface are implemented in Protocol Stack Pool. The system is implemented in Linux operating system and a web-based management subsystem is provided. To test the system, we set up an emulated core network, including SGSN emulator, GGSN emulator, Radius server and application server. We also access the current cellular mobile communications network. All procedures are verified correctly and applications can also run normally.

However, further work is still needed, including system optimization and more system testing. Whats more, we are now preparing to support testing of packet core network of LTE (Long Term Evolution) system. The related protocols are under development.

References

[1] http://openbsc.osmocom.org/.
[2] http://sourceforge.net/projects/ggsn/.
[3] 3GPP TS 23.060 V5.13.0 (2006-12) Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS); Service description; Stage 2(Release 5).
[4] 3GPP TS 24.008 V5.16.0 (2006-06). 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Mobile radio interface Layer 3 specification; Core network protocols; Stage 3 (Release 5) .
[5] Chai-Hien Gan, Yi-Bing Lin and Shi-Hi Chen. Design and implementation of UMTS session management in the user equipment. WIRELESS COMMUNICATIONS AND MOBILE COMPUTING, 2007: 755-766.
[6] 3GPP 29.060:General Packet Radio Service(GPRS); GPRS Tunnelling Protocol(GTP) across the Gn and Gp interface[S], 2006.
[7] 3GPP TS 25.415:UTRAN Iu Interface User Plane Protocols[S], 2006.
[8] 3GPP TS 25.412:UTRAN Iu Interface.Signalling Transport[S], 2006.
[9] http://savannah.nongnu.org/projects/lwip/.
[10] Framework Architecture for Signaling Transport[S]. IETF RFC2719, 1999.
[11] http://sourceforge.net/projects/lksctp.
[12] R. Stewart, Q. Xie, K. Momeault, T. Taylor,etc. Stream Control Transmission Protocol. RFC2960, IETF, October 2000.
[13] 3GPP TS 25.413:UTRAN Iu Interface. RANAP Signalling[S], 2006.
[14] G Sidebottom. Signaling System 7 (SS7) Message Transfer Part 3(MTP3)—User Adaptation Layer(M3UA) RFC3332 IETF Network Working Group-September 2002.

Xia Hailun is a lecturer at the School of Information and Communication Engineering of Beijing University of Posts and Telecommunications, China. He received his BEng and PhD degrees from BUPT-in the areas of Wireless Telecommunications in the years 2002 and 2007 respectively. His research field is in the areas of next generation wireless networks.