A Method for Testing the Protocol Stack of an Access Gateway to Broadband Network

Doris Bao[1, 2], Luca De Vito[1, 2], Laura Tomaciello[1, 2]
[1] *Dept. of Engineering, University of Sannio, Corso Garibaldi, 107, 82100 Benevento, Italy*
Ph.: +39 0824305600, Fax: +39 0824305840, E-mail: {doris.bao, devito, laura.tomaciello}@unisannio.it, http://lesim1.ing.unisannio.it
[2] *Telsey Telecommunications SpA, Benevento Research Laboratory, via dei Sanniti,1, 82018 San Giorgio del Sannio (BN), Italy.*
Ph.: +39 0824330145, Fax: +39 0824330145, http://www.telsey.com

Abstract **-** This paper describes the developement of a method to automate the validation phase of access gateways firmware releases. The method has been designed to stress the firmware in order to detect the software faults and minimize the possibility of device failures. Analysis about the performances of the proposed method shows its efficiency in term of execution time and reliability versus a manual investigation.

I. Introduction

During the last years the Digital Subscriber Line (DSL) technology continues to grow with the subscriber demand for broadband residential and business services. This is driven by the global demand for multiple applications, including new video and voice services that are allowed by high speed connections [1]. The access gateway is an integral part of the broadband network. It delivers multiple IP-based broadband services to customers' homes and offices over high speed, always-on, broadband connections. End users can access the most innovative and ground-breaking services through this device, which interconnects all peripherals, computers, television sets and telephones by using a single broadband connection.
Some researches [2] demonstrate that the significant change in technology over the last years is not only an increase in speed and reduction in power consumption, but also an improvement in the reliability of the devices. Hardware failure rates are currently decreasing while the relative contribution of firmware goes up. Telsey Telecommunications historical and statistical reports [3] show that access gateway faults are caused by the hardware in the 13% of the cases, by firmware problems in the 87% of the cases. Like an Operating System, the firmware runs on the access gateway to drive its hardware components and to implement its logical architecture. It is necessary to invest in firmware tests in order to ensure successful products. To cut down the cost of the testing phase, automated tests become more and more popular [4, 5]. Many instruments about automatic testing of access devices have been already developed. They are mainly based on protocol and technology conformance tests [6].
This work deals with the development and the implementation of a method to validate the interaction among the access gateway components that already passed the conformance tests. The developed method is able to test the correct behaviour of a firmware release, by stressing the communications between the hardware components and the network protocol layers. The proposed method is intended to test a Voice over Internet Protocol (VoIP) access gateway supporting H.323 protocol [7], anyway the designed method can be enlarged on all devices that provide access to broadband services. This paper is organized as follow: in Section II the access gateway protocol stack is described, in Section III the test method and its implementation is presented, the Section IV analyses the implemented tests and the way to perform them using the Graphical User Interface (GUI). Finally, in Section V, an analysis on the reliability and efficiency of the proposed method is reported.

II. Access Gateway Protocol Stack

An access gateway implements the Transport Control Protocol/Internet Protocol (TCP/IP) protocol stack specified in Fig.1 to provide users with broadband access. The physical layer handles the electrical and mechanical requirements necessary to transfer data between two nodes in a network. The gateway provides LAN access with different technologies: Ethernet, USB and 802.11 at the user side and with ADSL at the network side. Through the network layer the access gateway performs routing functionalities to forward packets on the LAN and WAN sides. In order to enable multiple hosts on the LAN side for accessing the Internet using a single public IP, the Network Address Translation (NAT) process involves re-writing the source and/or destination addresses of IP packets. The firewall permits or blocks applications in order to let them or forbid them to setup network connections. NAT and firewall functionalities are implemented on the network layer. At the application layer, the access gateway operates as a H.323 endpoint to manage VoIP communications. Moreover, TELetype NETwork (TELNET) access, with command line, and Hyper Text Transfer Protocol (HTTP) access,

with a graphical user interface by means of a WEB browser, allow local or remote management of the gateway, while Trivial File Transfer Protocol (TFTP) client enables the firmware updating process. The real operating scenario of the access gateway is also shown in Fig.1. The Digital Subscriber Line (DSL) architecture for high-speed Internet access is based on Asynchronous Transfer Mode (ATM). The ATM Permanent Virtual Connections (PVCs) provide connectivity from the access gateway device through the DSL Access Multiplexer (DSLAM) to the Broadband Remote Access Server (BRAS). The DSLAM links the user access DSL line to the data transport network, the BRAS authenticates the subscriber's credentials, validates the users access policies, and routes the data to their respective destinations on the Internet [8]. The access gateway is able to support different types of PVCs. In order to increase performance, often network providers make use of two different PVC, to deliver voice and data services separately. This is the case of the considered network architecture which uses a PVC with a IPoA (IP over ATM) protocol for voice traffic, and a second PVC with a PPPoA (PPP over ATM) protocol for data traffic. On the VoIP side, the access gateway supports the Registration, Admission and Status (RAS) protocol to communicate with the gatekeeper that serves as the central point for all calls within its zone of control.

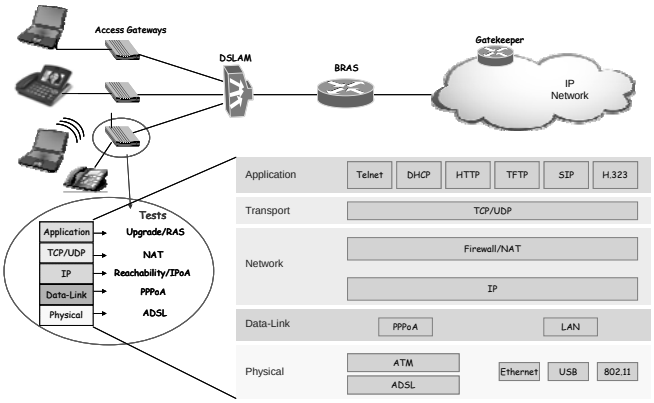

Fig.1. ADSL technology architecture and access gateway protocol stack.

III. Test Method

A. Basic Idea

The goal of the method is to put on trial, in automatic way, the firmware release of the access gateway, to detect the software faults and to minimize the possibility of device failures. This method is intended to be used in an alpha test phase. The firmware is evaluated using a black box approach: the method acts as an operator that does not access and does not know the source code. This leads to a proper and strict investigation. In literature, a series of pre-defined test cases for checking conformance protocol stack layers characteristics exist [6]. These tests generate fault events in a specific layer and analyse the correct communication of the layers of the communicating endpoints, which lie at same level of the network stack. Instead, the proposed method aims to analyze the interaction between contiguous layers located on the same endpoint (Fig.2). The idea of this method is to test, in automatic way, the behaviour and all functionalities of the access gateway when lower protocol stack layers change their state, so that the response of the upper layers can be verified. A configuration application running on the gateway is in charge of monitor all the interfaces, allowing to stop the functions of a layer when the lower ones are not running and to start them when the lower layers go up again. The method consists of substituting a real protocol stack layer with a simulated layer and to analyse the access gateway behaviour. This effort has to be repeated to test all the protocol stack layers.

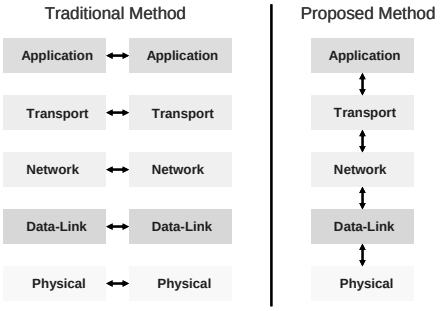

Fig.2. Traditional versus proposed method.

B. Method Implementation

The realized method has been implemented in a software tool running on a control unit which drives some hardware components. The scenario, shown in Fig.1 and Fig.3, acts as a sort of small Internet in laboratory and connects the control unit, a GNU/Linux device, to the access gateway through either LAN or the WAN interface (Fig.3) depending on the test. The two connections are different because the link from the control unit to the access gateway is different: in the first case it is an Ethernet cable, in the second case it is a copper pair carrying the ADSL signal. The software tool, running on the control unit, has been developed in C language. The implementation block scheme is reported in Fig.4. The tool accesses the gateway via TELNET, compels some test actions and checks the access gateway

behaviour. The tests cover all the basic functionalities and were chosen basing on of historical statistical motivations in order to put effort into common bugs. At the end a log file is generated, where the test date and the outcome are reported. Moreover, to help the user during the test a simple and intuitive Graphical User Interface (GUI) completes the software tool. Some buttons and text areas allow the operator the management of the whole test architecture, and to carry out the different tests (Fig.5). The GUI is divided in several sections: one for each test, one to configure the gateway, a display to show the test results, and an exit button that closes the control application. There is the possibility to execute some tests in series, too.

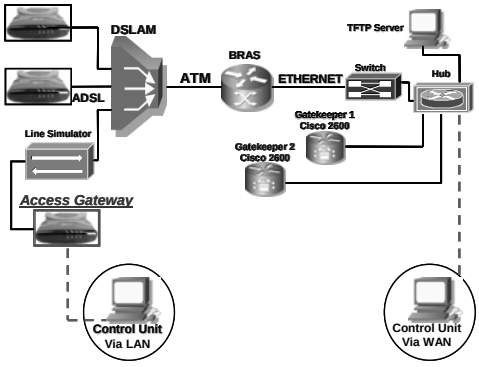

Fig.3. Test system components and Control Unit via LAN and via WAN.

Clicking on WAN ALL and LAN ALL button, the tests will be executed in series according to the following order: reachability test, download test, ADSL test, IPoA and PPPoA test, NAT and RAS test. When a test section starts, the first step is to configure the Access Gateway, setting the gateway IPoA and LAN addresses, and specifying the TFTP server IP address and the configuration file name, in the "Router Option" and "Router Configuration" sections, respectively. The two buttons, "LAN dwl" and "WAN dwl", force the gateway to download a new software image.

IV. Test Set

The following section details the implemented test and the GUI corresponding commands. In order to find out eventual bugs of the access gateway firmware, each test can be executed several time. The goal is to stress the firmware and recreate the conditions of repeatability of an individuated fault.

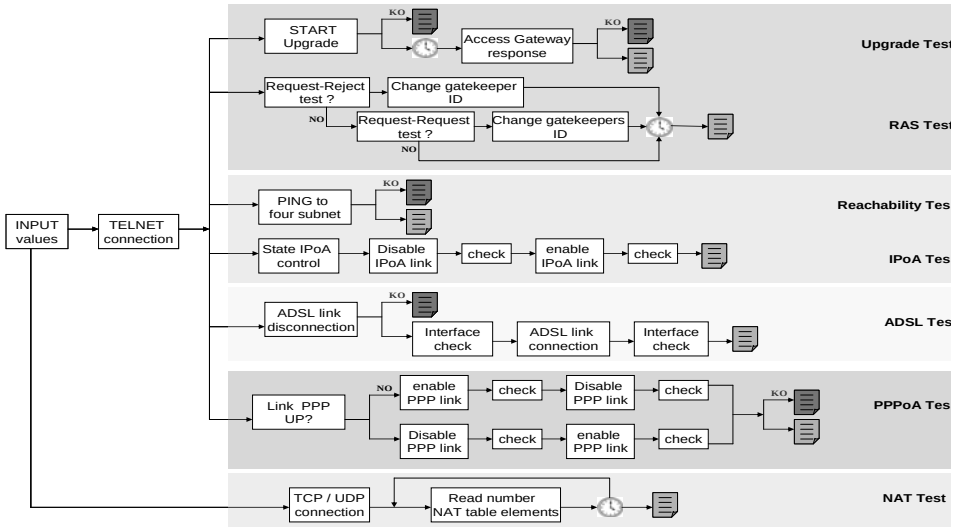

Fig.4. Implementation block scheme.

A. Upgrade Test

The firmware upgrade is necessary when new functionalities must be added to the access gateway or some faults have to be corrected. The release upgrade can be done via TFTP or HTTP. The *Upgrade Test* verifies if the upgrade process works correctly. The aim is to avoid a failure which causes the inaccessibility of the access gateway from remote and the support of an operator to the user's house to restore the blocked device. The implemented algorithm, (i) reads the input data such as the IPoA gateway address, the TFTP or HTTP server address, the firmware image name and the number of time to execute the test, (ii) opens a TELNET connection, (iii) starts the upgrade, (iv) wait for the upgrade time, and (v) finally save the results (Fig.4).

B. Reachability Test

The purpose of the *Reachability Test* is to determine whether the host is connected to a network on which it has a valid routable IP address. The test schedules to send pings to four representative hosts of four different subnets to which the gateway should physically be connected. If the ping arrives to the

destination IP address, the test gives a positive result (Fig.4). On the GUI, the *Reachability Test* section is composed of two buttons: LAN or WAN button. Clicking on LAN or WAN button of the GUI, the software tool sends pings to the Device Under Test (DUT) proper interface.

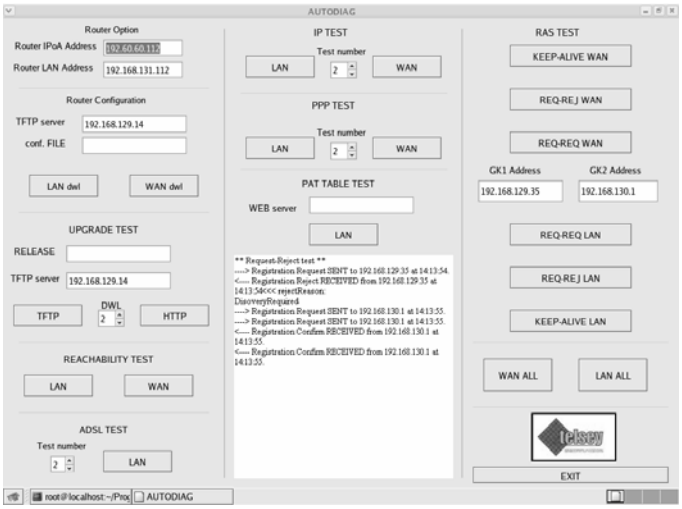

Fig.5. Graphical User Interface (GUI).

C. ADSL Test

The *ADSL Test* allows testing the gateway error resilience when it is subjected to frequent link connections and disconnections. Moreover, it executes a transversal gateway software debug verifying the PPPoA and IPoA interface state, and the communication between the physical level and the upper level protocols. The *ADSL Test* input parameters to be inserted in the GUI, are: LAN connection, IPoA address and number of times to execute the test. The automatic tool verifies the correct behaviour reaching the device via TELNET, displaying the gateway active interface table and verifying the link state.

D. NAT Table Test

The *NAT Table Test* verifies the NAT table capacity by counting the number of client connections that the device can manage. The function that performs this test accepts as input the LAN IP address of the client. During this procedure, the access gateway is connected to the test station via LAN. To verify that the router specifications are correct, several TCP and UDP connections are executed in groups of 100 until NAT table saturation. In order to start the *NAT Table Test*, the user has to click on LAN button.

E. RAS Test

RAS [9] is a part of the H.323 specifications that involves the addition of (or refusal to add) new authorized users, the admission of (or refusal to admit) authorized users based on available bandwidth, and the tracking of the status of all users. The function written for this test opens a RAS channel between the gatekeeper [10] and the terminals and verifies the requests that the device sends to gatekeeper. This section is divided in three different tests: (i) *Keep-alive Test*, (ii) *Request-Reject Test*, and (iii) *Request-Request Test*. The first one verifies the gateway registration keep-alive on the gatekeeper, sending a Registration ReQuest (RRQ) to the gatekeeper after a time chosen by the user. The second one controls the device behaviour when the gateway receives a Registration Reject from the primary gatekeeper and tries to register itself to the secondary gatekeeper. The last one analyses the router behaviour when the gateway registers itself neither to the primary nor to the secondary gatekeeper. On the user interface, the buttons, “KEEP-ALIVE WAN”, “REQ-REQ WAN” and “REQ-REJ WAN”, execute the test via WAN and other three button, “KEEP-ALIVE LAN”, “REQ-REQ LAN” and “REQ-REJ LAN”, start the test via LAN. The gatekeeper IP address has to be inserted by the user in the appropriate areas of the GUI.

F. IPoA Test

The *IPoA Test* does a limited number of interface connections and disconnections to test the link resilience. The goal of this test is to simulate the link loss and test the gateway ability to open a new IPoA session on the BRAS. The software tool accomplishes a TELNET connection, reads the gateway IP address assigned by the BRAS, and connects itself to the router via PPP. The *IPoA Test* starts clicking on LAN and WAN button and inserting the number of the test executions.

G. PPPoA Test

The *PPPoA Test* verifies the link state and the correct cancellation of Firewall and NAT tables after every PPP disconnection. The function in charge to test PPPoA link, first reads the input parameters and connects itself via TELNET, then it changes the link state. Every four connections and disconnections the link state (up or down) and Firewall and NAT tables are checked. The *PPPoA Test*

begins after the user chooses the iteration of the test and clicks on WAN or LAN button.

V. Validation of the Developed Method

The developed method was subjected to persuasive validation processes in order to assure the fidelity of the results returned by the tool in automatic way. In particular, some software releases with known bugs were analysed by the system and by manual investigation at the same time. The comparison, between the results of the automatic tests and the respective tests performed manually, demonstrated the total reliability of the developed system. Moreover, the system features significant advantages in terms of optimising effort and time involved in validation phase. In a manual analysis, these features are strictly tied to the operator's experience. Independently from the operator's skill, the automatic system makes easier to obtain and analyse the test results. The inherent benefits of the realized testing method include: easy of use, repeatability, and efficiency.

Ease of Use. The Graphical User Interface (Fig.5) is based on menus which allow users to perform testing without memorizing commands. The user can choose to perform the test individually if the analysis has to be conducted on a specific aspect or in series to obtain a complete debug of the access gateway software release under test. The user has always a feedback about the test results on the GUI display. Moreover, the system generates detailed log files for every test performed, which allow the user to trace the steps of the validation phase and to perform a simple and efficient analysis of the results, especially when a fault occurs.

Repeatability. The system gives huge advantage in the cases where the tests have to be run repeatedly. For example, when a test failure occurs, it may be necessary to repeat the test several times to demonstrate that the failure is not intermittent and to determine the exact fault that caused the failure. After the fault has been eliminated, it is then necessary to demonstrate that an identical test can be completed without the failure reoccurring. The realised testing method enhances the validation process by removing the element of human variability and errors when repeating tests.

Efficiency. The test set detailed in Section IV requires many repetitive tasks which computers can handle with speed and accuracy. Comparing computer aided testing to manual testing indicates that identical tests can be completed much more quickly using computer control. The system thus enhances the access gateway analysis process by reducing the time to perform required tasks and thereby reducing the total test interval. Running tests manually can be very time consuming.

A. Investigation of the Request-Reject Test

In order to show the advantages in the use of the realised system versus a manual investigation, in this section a study case on a RAS [10] *Request-Reject Test* is presented. The VoIP H.323 access gateway registers with the gatekeeper at startup. If it receives a reject from the primary gatekeeper the gateway sends another request to the secondary gatekeeper. This situation has to be recreated by the *Request-Reject Test.* A manual investigation and implementation of this test, schedules an access via TELNET or HTTP to the gateway in order to change the H.323 configuration to start a new *Registration Request* and cause the *Registration Reject* from the primary gatekeeper. The operator has to use a network protocol analyser, like Ethereal [11], to capture the signalling H.323 packets, in particular the H.225 packets exchanged between the access gateway and the gatekeepers. So the captured signalling flow has to be analysed by the operator to find out possible failures. This procedure means an inefficiency of time and the operator has to be expert of H.323 VoIP to catch the presence of errors. On the other side, using the realised system it is sufficient to configure "Router Option" and "Router Configuration" (Fig.5) on the GUI and to run the test clicking the appropriate button. In this case, the system accesses to the gateway via TELNET and performs all the necessary actions., this process is completely transparent to the operator. A deep knowledge of the H.323 is not required to the user, because the system analyses the signalling flow and reports the possible faults. Reading the test log file of the validation phase of a software release, an anomaly is noted, as shown:

*** Request-Reject test ***
----> Registration Request SENT to 192.168.129.35 at 14:13:54.
<---- Registration Reject RECEIVED from 192.168.129.35 at 14:13:54<<< rejectReason: Discovery Required
----> Registration Request SENT to 192.168.130.1 at 14:13:55.
----> Registration Request SENT to 192.168.130.1 at 14:13:55.
<---- Registration Confirm RECEIVED from 192.168.130.1 at 14:13:55.
<---- Registration Confirm RECEIVED from 192.168.130.1 at 14:13:55.

The device delivers a registration request to the primary gatekeeper but receives a reject, so tries to contact the secondary gatekeeper but it sends two requests at the same time. It is an incorrect behaviour of the device and the validation automatic tool detects this firmware fault.

In conclusion, performing the *Request-Reject Test* on the same software release, a manual analysis

requires about 610s and repeating the test more times with different operators only in the 80% of the cases the fault was correctly detected. The automatic system spends 450s to perform the test and individuates always the fault.

B. Advantages of the Developed System

The same analysis detailed for the *Request-Reject Test* was conducted for the whole test set to obtain the comparison shown in Tab.1. A significant number of access gateway software releases, with known bugs, have been validated using the developed method and by a manual investigation. The table reports the average results in term of execution time, and percentage of success that is the correct behaviour in the characterization of the failures. It can be easily seen that the proposed method gives many advantages in terms of time effort versus manual testing. The reliability and the efficiency of the implemented method shows the efficacy of the performed research.

Table.1: Performances evaluation of the manual analysis versus the use of the developed method.

	MANUAL				**AUTOMATIC**			
	Execution Time		**Success Rate**		**Execution Time**		**Success Rate**	
	LAN	**WAN**	**LAN**	**WAN**	**LAN**	**WAN**	**LAN**	**WAN**
Upgrade Test								
- HTTP	135s	137s	100%	100%	88s	91s	100%	100%
- TFTP	195s	200s	100%	100%	120s	127s	100%	100%
Reachability Test	87s	90s	100%	100%	32s	34s	100%	100%
ADSL Test	218s	-	100%	-	75s	-	100%	-
NAT Table Test	340s	-	100%	-	190s	-	100%	-
RAS Tests								
- Keep-alive	604s	606s	100%	100%	445s	446s	100%	100%
- Request-Reject	608s	610s	80%	80%	447s	450s	100%	100%
- RequestRequest	169s	173s	100%	100%	80s	83s	100%	100%
IPoA Test	420s	424s	100%	100%	380s	377s	100%	100%
PPPoA Test	315s	318s	100%	100%	282s	285s	100%	100%

VI. Conclusions

The paper deals with the characterization of a method to automate the validation phase of an access gateway firmware release. The proposed method aims to put on trial the firmware release of the access gateway, to detect the software faults and minimize the possibility of device failures. A method capable to stress the firmware and recreate the conditions of repeatability of an individuated fault has been implemented. The analysis about the performances of the proposed method shows its reliability and efficiency.

References

[1] www.dslforum.org.

[2] D.P.Siewiorek, R.Chillarege, T. Kalbarczyk, “Reflections on industry trends and experimental research in dependability”, *IEEE Transactions on Dependable and Secure Computing*, 2004, pp.109-127.

[3] Telsey telecommunications S.p.A., www.telsey.com.

[4] P.Daponte, D.Grimaldi, M.Marinov, “Real-time measurement and control of an industrial system over a standard network: Implementation of a prototype for educational purposes”, *IEEE Trans. on Instrum. and Meas.*, vol. 51, No.5, Oct. 2002, pp.962-969.

[5] L.De Vito, S.Rapuano, L.Tomaciello, “One-Way Delay measurement: state of art”, *Proc. of IEEE IMTC/2006*, Sorrento (NA), Italy, 24-27 Apr., 2006, pp.218-223.

[6] Spirent Communications, www.spirentcom.com/documents/(380.pdf, 3195.pdf).

[7] G.A.Thom, "H.323: the multimedia communications standard for local area networks", *IEEE Communications Magazine*, vol. 34, Dec. 1996, pp. 52-56.

[8] Agilent Technologies, “Understanding DSLAM and B-RAS access device”, 2006 http://cp.literature.agilent.com.

[9] “Voice over IP: Protocols and Standards”, http://www.cis.ohio-state.edu/~jain/cis788-99/voip_protocols/index.html.

[10] Cisco System, “Understanding H323 gatekeepers”, http://www.cisco.com.

[11] Ethereal: A Network Protocol Analyzer, http://www.ethereal.com/.