GPRS FUNCTION TEST USING TTCN

Endre HORVÁTH* and Axel MANTHEY**

*Department of Telecommunication and Telematics,
Budapest University of Technology and Economics,
1117 Budapest, Pázmány Péter sétány 1/D, Hungary,
e-mail: horvathe@ttt-atm.ttt.bme.hu
**Ericsson Eurolab Deutschland GmbH,
Ericssonalle 1, D-52134 Herzogenrath, Germany,
e-mail: Axel.Manthey@eed.ericsson.se

Received: August 31, 2000

Abstract

Functional and conformance testing of new standardized services and protocols is a challenging task as standards may change during implementation and the test system has to be adapted to the system under test continuously. This paper describes how modular and concurrent TTCN was deployed for validation of the GPRS support nodes and reports on the experience collected.

Keywords: GPRS, TTCN, functional testing.

1. Introduction

During the last few years the number and complexity of applied telecom and datacom protocols have increased. The introduction of General Packet Radio Services is connected with the challenge to carry out functional and conformance tests. Never before such a large number of telecom and datacom protocols have been combined in a single network element. Demands to the test system are high: the system should be able to grow with the implementation, support a large variety of protocols in different national variants and be utilized in simulated and target environment. Section 2 gives a brief introduction to Conformance and Functional Testing and TTCN. In section 3 there is a short summary of General Packet Radio Services presenting the interfaces and protocols that need to be tested. The chosen solution is described in Section 4 emphasizing the deployment of concurrent and modular TTCN. Section 5 summarizes the experience collected throughout the testing project.

2. Functional Testing vs. Conformance Testing

The general aim of Conformance and Functional Testing is to prove the conformity of an implementation according to standards, but it is achieved in different design phases.

Conformance testing examines the final products and represents an abstract level of verification. The purpose of conformance testing is to increase the probability that different OSI implementations are able to interact [8][10]. Functional testing is part of the production and its purpose is to verify the functionality of a Software issue. It follows the design flow (phases) and verifies the currently implemented functionality.

Another main difference between Functional Testing and Conformance Testing is who benefits mainly from the tests. Conformance Testing provides information (comparable test results) for telecom suppliers and network operators about the conformity of products to the standards and their interoperability with other implementations. Functional Testing helps developers/manufacturers to find and correct errors in order to increase the quality of the products and it facilitates consistent regression testing during design.

2.1. TTCN Language

TTCN (Tree and Tabular Combined Notation) is an internationally standardized test notation designed for testing of protocol implementations based on the OSI (Open System Interconnection) Reference Model [8]. TTCN is independent of test methods, layers and protocols and reflects the abstract testing methodology defined in ISO/IEC 9646-1 and ISO/IEC 9646-2. However, TTCN was originally designed for conformance testing of telecommunication protocols, its generality and flexibility make it a powerful notation in other test situations as well: it can be used for various testing purposes through the entire development process of telecom and datacom products.

The TTCN language standard specifies two forms of notation: a human-readable graphical form (TTCN.GR) used for test suites' production and a machine processable form (TTCN.MP) suitable for processing within and between computer systems (e.g. test tools) [8]. These two forms are semantically equivalent.

A TTCN test suite is divided into four parts [8]. An overview generated from the other three, a declaration part where all messages are declared, a constraint part where instances/variables are constructed based on the declarations and a dynamic part consisting of the actual test cases. The test cases may be assorted to test groups. Each test case consists of sequences of events that are initiated or observed at the test interfaces and contain all the information necessary to fully specify the test purpose in terms of the protocol that the IUT (Implementation Under Test) is supposed to implement. The specification of those test cases in which more than one behaviour description is to run simultaneously is handled by the concurrency features, particularly involving the definition of Parallel Test Components and Test Component Configurations [10]. TTCN presents all definitions and dynamic actions (behaviour trees) in tabular format (*Fig.* 1). The basic structure of the test suite and its tables are unambiguously defined.

The TTCN notation is a generic test notation, meaning that the language could

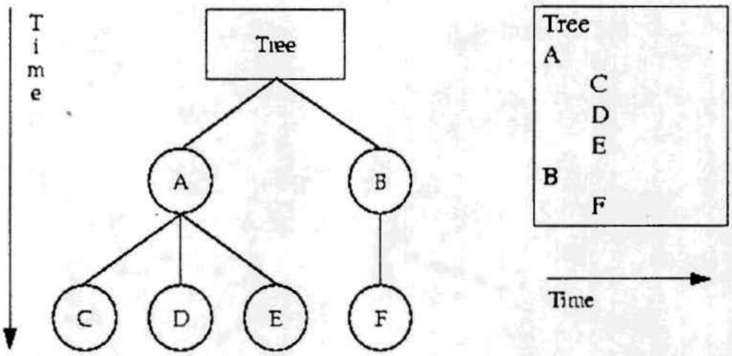

Fig. 1 Realization of behaviour trees in tabular format

be applied in testing of any protocol level of a telecommunication interface. This makes it necessary to identify, from an abstract level, where the interaction with the test object should take place. The interaction point is called PCO - Point of Control and Observation. All interactions with the test object are performed via one or several PCOs defined in the test suite [10]. Every PCO has one or several ASPs, Abstract Service Primitives connected to it. The ASPs define the methods of how to interact with the test object. Since TTCN is originally designed for the purpose of conformance testing, the ASPs correlate to the primitives defined in the standards of the protocol tested. The ASPs often convey PDUs, Protocol Data Units that are the messages between the peer protocol layers. Basic events in TTCN are sending and receiving PDUs through the appropriate PCOs by the help of ASPs and the timer events. So the data structure of the test events is multilayered: PDUs are embedded into ASPs and both PDUs and ASPs may contain structured fields.

TTCN provides good logical structuring of test suite elements (data and events) and support complex data structures including usage of ASN.1 (Abstract Syntax Notation One) [9], moreover TTCN modules are defined to allow sharing of commonly used TTCN specifications (definitions, constraints, test cases/steps) between test suites supporting reuse of TTCN code.

3. General Packet Radio Services

General Packet Radio Services (GPRS) is a standardized extension to existing GSM networks that offers packet switched data services. Two new network elements will be added to the GSM network architecture: the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN) [2]. These nodes are interconnected by means of an IP based core network and have signaling connections to existing GSM network elements such as Home Location Registers (HLR), Mobile Services Switching Center/Visitor Location Registers (MSC/VLR), Base Station Controllers (BSC) or Short Message Service Gateway MSCs and Interworking

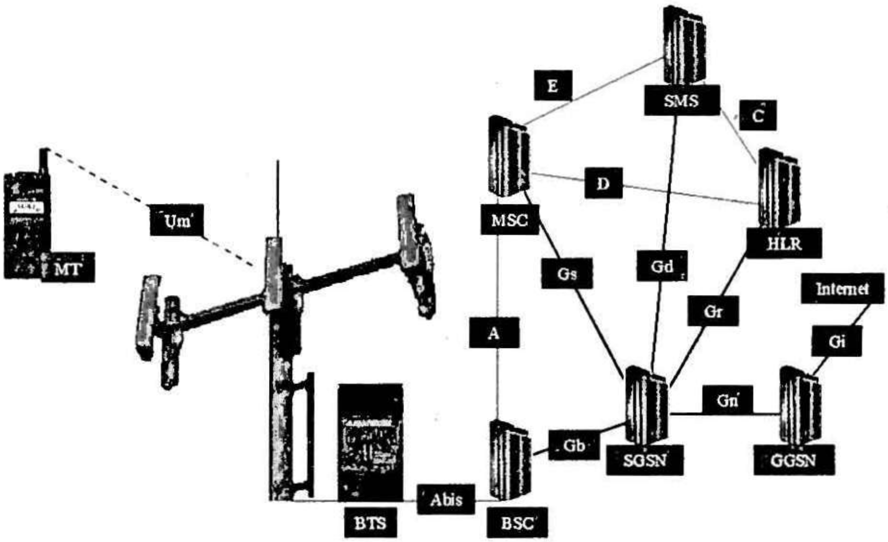

Fig. 2 Extension of the GSM network: new GPRS nodes and interfaces

MSCs (SMS-GMSC, SMS-IWMSC) (*Fig. 2*). The enhancement of the GSM network requires software updates in the HLR and MSC/VLR, hard- and software updates in the BSC and the design of both new nodes SGSN/GGSN. The SGSN serves packet data users in a given geographical area while the GGSN connects to external packet data networks.

From an end user's perspective GPRS offers permanent connectivity to IP networks, volume based charging and a higher bandwidth compared to existing GSM data services (up to 115 Kbps). Circuit switched GSM services and GPRS can coexist without disturbances, only one HLR based subscription is needed [2]. Radio resources can be shared efficiently among several users. Horizontal applications (e.g. e-mail, FTP, HTTP) and vertical applications (such as telemetry, diagnostics, vending machines) can be offered. In addition SMS will be supported over GPRS [1].

To connect the SGSN and GGSN to existing GSM network elements new protocols were standardized and existing protocols were modified to support the new services (*Fig.* 3). The GPRS Tunneling Protocol (GTP) transports signaling and payload between GSNs [7]. The User Datagram Protocol (UDP) over IP is the bearer for GTP Protocol Data Units (PDU). HLR, SMS-GMSC, SMS-IWMSC and Equipment Identity Register (EIR) use Mobile Application Part (MAP) [6] PDUs over Transaction Capability Application Part (TCAP). The MSC/VLR is connected via the Base Station System Application Part + (BSSAP+) protocol using Signaling Connection Control Part (SCCP) Abstract Service Primitives (ASP). The interface to the BSC uses GPRS Mobility Management (GMM) [3], Session Management (SM) and SMS PDUs over Logical Link Control (LLC) [4] ASPs on the signaling

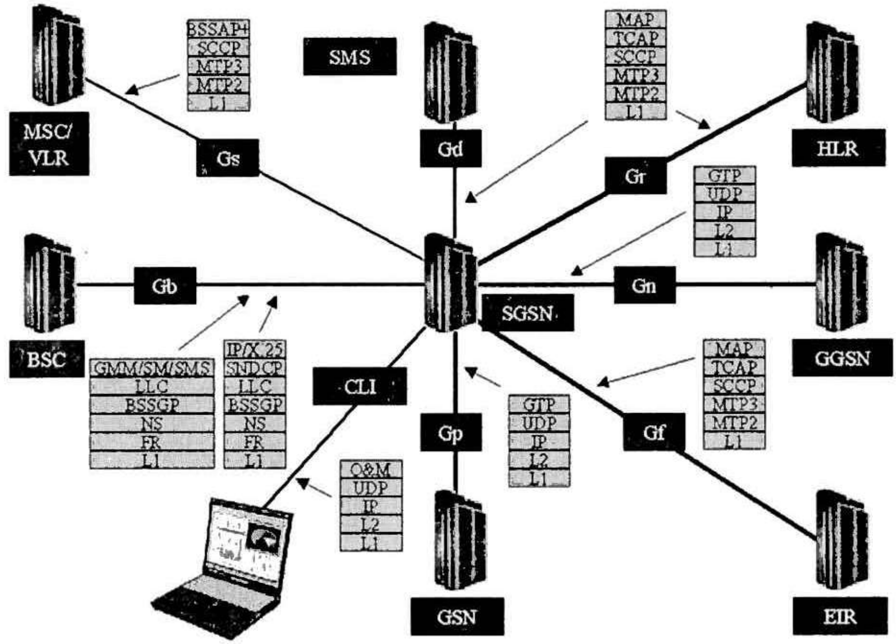

Fig. 3 Interfaces to the SGSN and involved protocols

plane and the Subnetwork Dependent Convergence Protocol (SNDCP) [5] over LLC for the transmission of packet data. For configuration and monitoring purposes a Command Line Interface (CLI) exists in addition to graphical user interfaces.

All these interfaces to the GSNs have to be simulated in order to test the newly implemented protocols, therefore testing a GPRS support node requires a complex test configuration simulating the other nodes adjacent to this entity. Furthermore, the protocols are affected by changes in standardization during design and not all the interfaces are available from the first design increment on: the test system is becoming more and more sophisticated as the design grows, thus the test system has to be upgraded permanently. These expectations can be met by using Concurrent and Modular TTCN test suites.

4. Functional Testing with TTCN

4.1. Usage of Modular TTCN

Modular TTCN facilitates work-distribution (parallel work) and it increases the reusability of the test suites. It is the best possible way to handle large sized test suite production.

The key to modular test suite production is the architecture of modularization.

In the literature there are three main modularization levels: Functional, Organizational and Language level.

Functional level modularization describes the partition of test suites into several functional parts, based on the tree structure of the test cases/groups. Organizational level modularization is set up according to the ATS (Abstract Test Suite) writer's point of view. This level allows a design where modules are shared between ATSs (Global Module, Common Module, Specific Module). Language level modularization is based on the classification of TTCN objects by the parts or sub-parts they are belonging to (i.e.: constants, test suite variables, PCOs, Timers, ASPs, PDUs, etc.).

The GPRS project is based on a combination of the above mentioned levels. Protocol definitions are located in separate modules. Commonly used test steps and constraints are dealt within separate modules according to the different protocols/network elements to be simulated. The MAP protocol definitions are extracted from the ASN.1 definitions of GSM TS 09.02 utilizing the possibility to import ASN.1 definitions in TTCN test suites [8][9]. *Fig. 4* shows the architecture of modularization applied in GPRS function test.

It is important to realize that the test suite production within the GPRS project was carried out in two steps: The first part was to develop a test suite template, i.e. to prepare the modules containing the protocol definitions, the common constraints and test steps. The second part was to build the actual test suite including the test cases. Tested protocols and functions are divided into test objects and a test suite is built for each of them using the modules of the test suite template. The test cases are located in these modular test suites, named according to the scope of the different test objects and importing the needed definitions from the connected modules (*Fig. 4*). In this way each tester is able to work on his own test module having the appropriate import declarations from other modules.

The protocol modules are updated centrally and as more protocols are implemented in design, those can be added to the test environment in the same incremental approach. To handle the versions and to store the different modules the well-known version-control system ClearCase was used. Via this system each user has his own view (virtual workspace), which enables him to work on the files in ClearCase as if he was in regular Unix directories with regular tools on regular files.

4.2. Usage of Concurrent TTCN

There are many network elements that need to be connected to the Implementation Under Test (IUT), the SGSN or GGSN in this project. Concurrent TTCN offers the possibility to handle the parallel communication in an efficient way [8][10].

Connected network elements are simulated by using parallel test components (PTCs) with Points of Control and Observation (PCO) associated. All parallel test components are controlled by one master test component (MTC). The MTC controls and synchronizes the PTCs: the body part of the test cases consists exclusively of

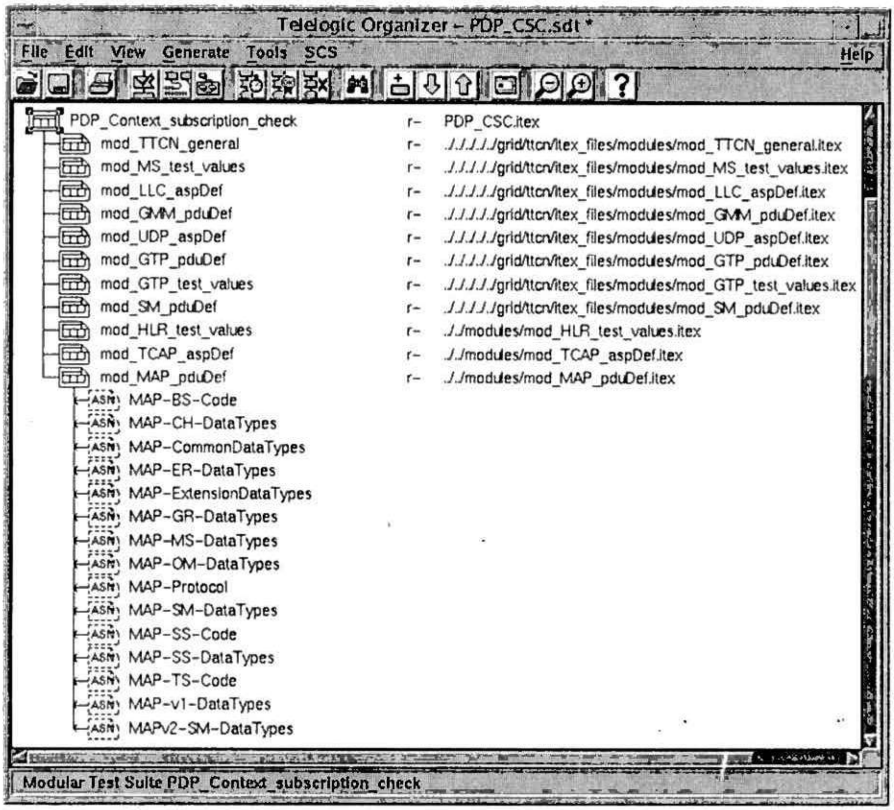

Fig. 4 Architecture for modularization

CREATE/ ?DONE statement pairs controlling the parallel test components and assigning local behavior trees (local steps) to each PTC (*Fig. 6*). Advantages of this strategy are that the test cases are easy to read and the configuration of new interface types is very simple. It is not difficult to follow the signaling on the different interfaces using local behavior trees for the PTCs. In order to avoid too long test cases, test steps are used for logically connected events to simplify their structure (*Fig. 5*) and to increase the re-usage of TTCN code.

Each PTC described in the test cases is tagged with a qualifier containing a test suite parameter (*Fig. 5*). This method allows enabling or disabling of the behavior trees: simulated network elements can easily be added or removed by using test suite parameters. When the network configuration changes (e.g. simulated/target environment), the test steps and constraints used in the test cases do not need to be changed - only the test suite parameters in the test suite configuration file have to be updated. This facilitates the partial replacement of simulated network elements by the real environment.

The parameterization of the test suites has a remarkable effect on reusability

Nr	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		+preamble			
2		CREATE (MS_PTC:local_MS_tree, HLR_PTC:local_HLR_tree, GGSN_PTC:local_GGSN_tree)			
3		?DONE (MS_PTC, HLR_PTC, GGSN_PTC)			
4		+postamble			
		local_MS_tree			
5		[TSP_simulate_MS]			
6		+TS_gmm_preamble			
7		+TS_gmm_Attach_req_acc_comp_imsi_gprs			
8		+TS_Activate_PDP_Context			NSAPI = 6 TI = 0
9		+TS_MS_postamble			
10		[NOT TSP_simulate_MS]			
		local_HLR_tree			
11		[TSP_simulate_HLR]			
12		+Successful_Fetching_of_Authentication_Triplets			
13		+TS_map_Successful_Update_GPRS_Location_pdpContext_par (bc_map_PDP_context_list)			
14		[NOT TSP_simulate_HLR]			
		local_GGSN_tree			
15		[TSP_simulate_GGSN]			
16		+TS_ggsn_Create_PDP_Context			
17		+GTP_postamble			
18		[NOT TSP_simulate_GGSN]			

Fig. 5 Test case example: usage of Concurrent TTCN

and adaptability. Not only network configuration aspects can be parameterized in the test suites: address information (e.g. node addresses), subscriber data (e.g. International Mobile Subscriber Identity, IMSI) in constraints and other configuration related data such as execution control parameters/qualifiers are handled by using test suite parameters. The proper and reasonable usage of constraints together with test suite parameterization ensures a good portability, reusability and maintainability of test suites.

4.3. An Example: Testing of Security Functions

The purpose of the security function is to prevent unauthorised GPRS service usage, to provide user identity and to ensure user data confidentiality. The security function is handled by the authentication and the ciphering procedures [2].

The aim of the authentication procedure is to protect the SGSN (Serving GPRS Support Node) against unauthorised use. The authentication procedure performs the identification and authentication of the service requester as well as carries out the validation of the service request type to ensure that the user is authorised to use the particular network service. Those authentication procedures shall be used that have already been defined in GSM, with the distinction that the procedures will be executed by the SGSN.

Some signalling information elements are considered as sensitive and must be protected. The confidentiality of user information concerns the information transmitted on the logical connection between MS (Mobile Station) and SGSN. These requirements for a protected mode of transmission are fulfilled by a ciphering function within the LLC (Logical Link Control) layer [4].

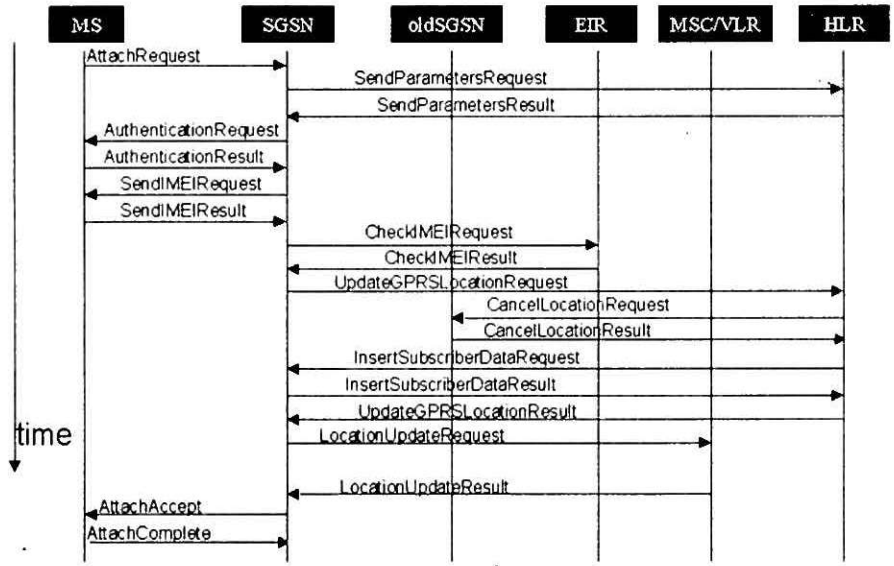

Fig. 6 Message flow of a successful Attach

Due to an Attach Request message sent by the Mobile Station (*Fig. 6*), the SGSN shall perform the authentication procedure. If the SGSN does not have previously stored (unused) authentication triplets (RAND: random number; SRES: signed response, calculated from RAND and IMSI; Kc: ciphering key) a request for new triplets (Send Authentication Info) will be sent to the HLR. The HLR responds with Send Authentication Info Ack containing the triplets. After that the SGSN sends Authentication and Ciphering Request message (with the random number) to enable the MS to calculate the SRES and the ciphering key. The MS responds with the Authentication and Ciphering Response message conveying the calculated SRES. Then the SRES stored in SGSN is compared with the SRES included in the Authentication and Ciphering Response. If the two SRES values are equal, the authentication is successful and ciphering can be started using the ciphering key (Kc). *Fig. 7* shows a test case example used for testing of a successful authentication.

Nr	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		+preamble			
2		CREATE(MS_PTC:local_MS_tree, HLR_PTC:local_HLR_tree)			
3		? DONE (MS_PTC, HLR_PTC)			
4		+postamble			collects verdict
		local_MS_tree			
5		[TSP_simulate_MS]			
6		+TS_gmm_preamble			
7		+TS_BSC_new_cell_id (TSC_gmm_RA1_Cell1)			
8		+TS_gmm_Attach_req_imsi_gprs_no_AC			
9		+TS_gmm_Authentication_and_ciphering_request			
10		+TS_gmm_Authentication_and_ciphering_response			
11		+TS_gmm_Attach_acc_imsi_gprs			
12		+TS_gmm_Attach_comp_imsi_gprs			
13		+TS_gmm_postamble			
14		[NOT TSP_simulate_MS]			
		local_HLR_tree			
15		[TSP_simulate_HLR]			
16		+Successful_Fetching_of_Authentication_Triplets			
17		+Successful_Update_GPRS_Location			
18		[NOT TSP_simulate_HLR]			

Fig. 7 TTCN test case for testing of Security functions

4.4. Test Configuration

The Abstract Test Configuration (ATC) is the representation of the testing environment within the TTCN test suites. It contains a description about how the test events have to be carried out by means of interfaces and communicating channels. Within the test suites the ATC is specified on a logical level and the executing environment interprets it towards the IUT by the help of parallel communicating programs representing the test components.

The executing tool for GPRS function test is the System Certification System (SCS) developed by Ericsson. The SCS is a set of test tools developed for executing test suites (given in standardized TTCN.MP format) towards the system under test (SUT). SCS offers the test management system that controls the configuration as well as the execution of the TTCN test suites. The results of the test execution are stored in log files.

The SCS tools have an open architecture, which allows to adapt the core system to any protocol or other well-defined interface. The interface adapters – called test ports – are dynamically linked to the SCS when the test session is started. For each type of interfaces a test port has to be developed.

The test port is liable for the communication between the SCS and the interface of the implementation under test (IUT). A test port has to be assigned to each PCO in order to execute the test suites. The assigned test port has to offer exactly the same set of ASPs that are specified in the test suite for the given PCO, e.g. LLC, TCAP, UDP or SCCP ASPs that are provided for the higher protocol layers. *Fig. 8* shows the test configuration used for testing the SGSN with appropriate test ports applied on different interfaces.

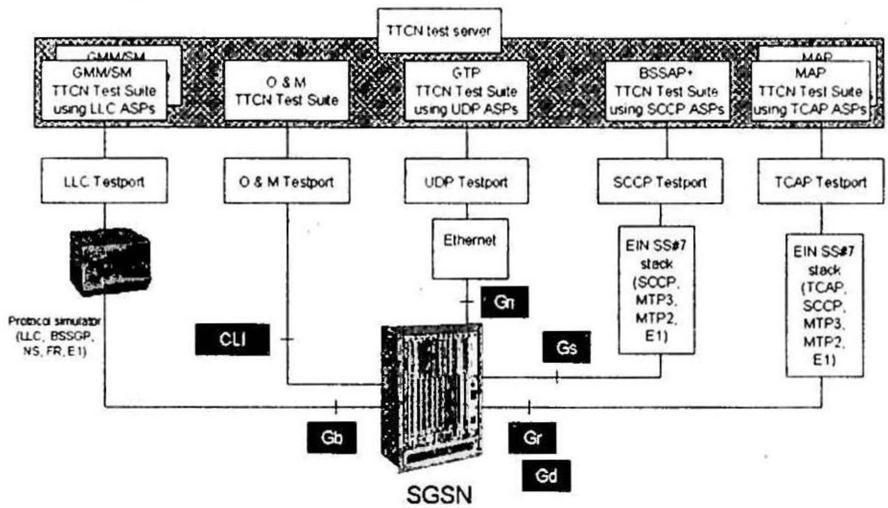

Fig. 8 Test configuration for testing the SGSN

Each test port in the SGSN test configuration has a direct connection to the IUT via real protocol stacks except for the LLC test port (*Fig. 8*): the BSS (Base Station Subsystem) is simulated by a protocol simulator. Besides using the complete protocol stack it is also possible to interface the IUT on a higher protocol layer. This example shows the flexibility of the test port concept in the SCS: the test suite can be executed in simulated as well as in target environment.

5. Conclusions

TTCN test suites are independent of test methods, layers, protocols or test tools, so TTCN is applicable under many conditions. It enables the handling of different protocols and multiple interfaces within one test suite and supports testing of complicated systems like the GPRS support nodes combining telecom and datacom protocols. TTCN provides powerful means of modularization and parameterization of test suites.

A TTCN based test system can provide homogeneous test environment for all protocols and interfaces just by using suitable test suites and test ports made for certain test purposes. TTCN test notation combined with a flexible and powerful execution platform - such as SCS - can increase the efficiency of telecom and datacom protocols' testing to a completely new level. At the same time the maintainability of test scripts and automation of testing can be considerably improved.

Writing of TTCN test suites is in many aspects similar to software development. A well-structured setup paves the way to a flexible and maintainable test system. A huge amount of tables in the test suites and a lot of codes and definitions need to be written before testing can start. TTCN offers a lot of flexibility that is connected with the risk of making mistakes. Central configuration management is essential for success, without it the parallel testing of different protocol versions would be nearly impossible. A well-configured executing environment provides a lot of possibilities: interactive, automatic and distributed test execution, simulation of non-existent components and excellent logging/tracing facilities.

A total of approximately 2500 test cases distributed over ca. 50 test objects has been produced and executed during the GSN verification project using TTCN and SCS. A large subset of these test cases has been repeated in all design increments (regression test) and helped to secure the product quality with a minimum of test case redesign.

References

[1] GSM 03.40, Digital cellular telecommunications system (Phase 2+); Technical realization of the Short Message Service (SMS); Point-to-Point (PP)

[2] GSM 03.60, Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Service description; Stage 2

[3] GSM 04.08, Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Mobile radio interface layer 3 specification

[4] GSM 04.64, Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Logical Link Control (LLC)

[5] GSM 04.65, Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Subnetwork Dependent Convergence Protocol (SNDCP)

[6] GSM 09.02, Digital cellular telecommunications system (Phase 2+); Mobile Application Part (MAP) specification

[7] GSM 09.60, Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp Interface

[8] ISO/IEC 9646, OSI Conformance Testing Methodology and Framework

[9] ISO/IEC 8824, Abstract Syntax Notation One

[10] BAUMGARTEN, B.–GIESSLER, A., OSI Conformance Testing Methodology and TTCN, North holland, 1994