J. Indian Inst. Sci., March–April 2000, **80**, 253–271.

Wireless application protocol (WAP) – State of the art

KATALIN TARNAY[1, 2], BARNABAS GOGOS[2], CSABA VILMOS ROTTER[1]
[1]Nokia Wireless Software Solutions and [2]University of Veszprem

Abstract

The paper introduces an up-to-date wireless solution, the WAP (wireless application protocol). After a short overview, WAP architecture is discussed explaining the WAP protocol hierarchy. One protocol, the WTP (wireless transaction protocol) was selected to demonstrate the main features of WAP protocols. Some new applications based on WAP are presented.

Keywords: Wireless application protocol and conformance testing.

1. Introduction

The Wireless Application Protocol (WAP) is an open, global standard empowering users of mobile phones and wireless devices to access and instantly interact with Internet information and communication services. The access to wireline Internet is secure. WAP is a major step in building the wireless Internet. A strategic forecast indicates more than 530 million wireless subscribers by 2001. A substantial portion of the phones sold in the near future will have multimedia capabilities, which include the ability to retrieve e-mail, and push-and-pull information from the Internet.

The wireless telecommunication industry leaders, Nokia, Motorola, Ericsson and Unwired Planet founded the WAP Forum in June 1997 (www.wapforum.org). The goal of the Forum is to develop technical standards for WAP protocols and to ensure Internet communications coupled with advanced telephony services. High-level technical standards and wireless application interoperability is a good base for WAP-enabled services and applications. The potential market for WAP services includes today's 300 million users of wireless phones and services, which is expected to double during the next years.

We intend to present in this paper the main technical features of WAP. The paper is structured as follows. First, we give an overview on WAP indicating its objectives and the advantages that accrue to end-users, telecom operators and service providers. The second part deals with WAP architecture and protocols. The third part considers a WAP protocol, the WTP (Wireless Transaction Protocol) in detail. The formal specification methods will be discussed in Section 4. Testing ensures reliability and high quality of WAP software, and hence, we discuss two faces of testing. Software testing is explained in Section 5 and conformance testing in Section 6. Finally, a collection of WAP applications is summarized in Section 7.

1.1. *WAP overview*

Providing Internet and web-based services on a wireless data network presents many challenges to service providers, application developers and handset manufacturers. Ensuring interoperability is important. Both bearer and device independence help foster interoperability. Each WAP component will communicate with all other components in the network by using the standard methods and protocols defined in the specification.

The main objectives of WAP are the following:

- to enable develop new applications that run on a mobile terminal
- to define an end-to-end application protocol that allows mobile terminal communicating with a server application
- guarantees interoperability among different terminals and servers
- implements end-to-end security
- defines an application environment that allows easy construction of end-to-end applications where the client part is downloadable
- makes client's application run on any mobile terminal
- protects the terminal from hostile applications
- provides connectivity with and an evolution path to the Internet

The main features of WAP are given in Fig. 1.

WAP is a good answer to the above-enumerated objectives. Industrial groups gain substantially from WAP solutions but end-users, telecom operators and service providers also have significant benefits in using WAP.

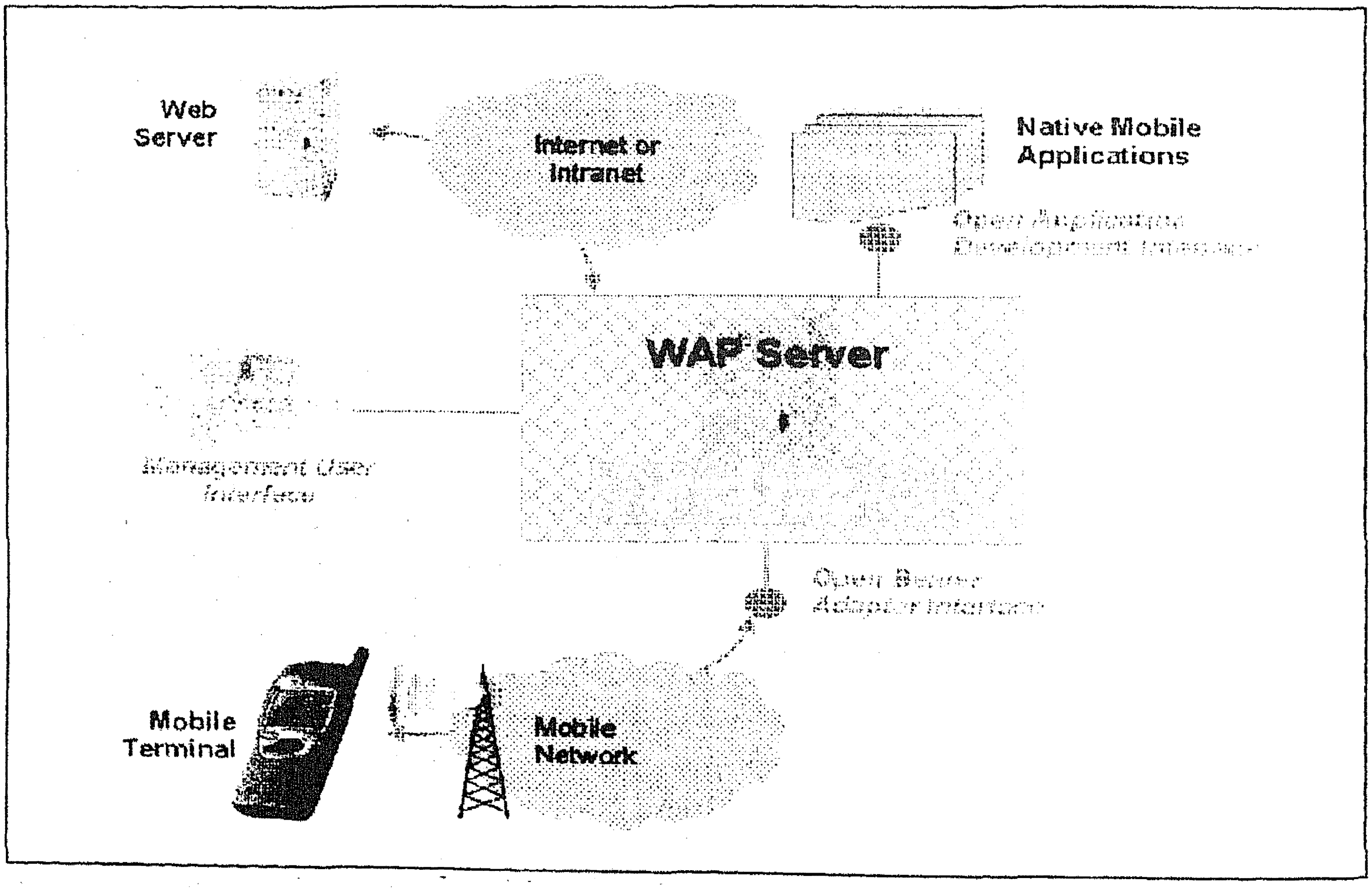

Web Server
Internet or Intranet
Native Mobile Applications
Open Application Development Interface
WAP Server
Management User Interface
Open Bearer Adaptor Interface
Mobile Terminal
Mobile Network

FIG. 1. Architectural view of WAP network.

The benefits for end-users are:

- instant access to any information via mobile handset
- mobile electronic commerce
- news and entertainment services
- highly secure communication when needed
- a wireless internet in pocket

The benefits for telecom operators are:

- more network traffic, more revenue
- same applications and services to several terminal types
- more access to mobile station enabling versatile applications

The benefits for service providers are:

- hundreds of millions of new customers to existing and new services
- generic services such as Internet information access
- database retrieval
- business domain-specific services

It is worth mentioning that WAP offers independent software vendors and system integrators new business opportunity, usage of tools and programming methods familiar from the Internet.

2. WAP architecture

2.1. *Background*

WAP architectural structure is developed and maintained by WAP Forum. Architectural specification of WAP[1] contains not only questions about the protocol stack and the WAP communication model, but also the motivation of development, the goals and the features.

2.1.1. *Motivation and constraints*

The main purpose of WAP is to combine the two emerging technologies, the Internet and wireless networking. Naturally, WAP-capable mobile devices offer not only the usual Internet and wireless services but also a combination of them. Thus, this system will provide new opportunities that have a great influence on both the technologies; for example, call control with customized user interface or mobile access to common Internet services like World Wide Web.

However, the majority of Internet services are developed for desktop computers that have enormous resource capabilities compared to hand-held mobile devices, which will be the client side of WAP services. Hence, the WAP architectural designer should take into account several constraints such as:

- less CPU capacity,
- less memory,

- restricted power consumption,
- much smaller displays, and
- different input devices.

The wireless network's infrastructure has been designed for telephone service and the WAP technology designer has to cope with

- less bandwidth,
- large and variable delay,
- unstable connections, and
- less predictable availability.

2.1.2. *Requirements*

WAP protocols and applications have to take into account the constraints mentioned above and the desired features. To satisfy all the requirements, WAP has to offer the following features:

- Interoperability — connect terminals of different manufacturers.
- Scalability — operate on several wireless networks and be able to use both limited and advanced services.
- Efficiency — provide suitable quality of service depending on the given mobile network.
- Reliability — provide a consistent and predictable platform for deploying services and applications
- Security — assure the integrity and safety of user data.

2.2. *Architecture overview*

2.2.1. *The Web model*

The major part of the Internet traffic is world wide web (www), so it is useful to compare the WAP model to the web. The www communication is based on client–server architecture. Web client applications, the so-called browsers, seek services from www servers. The given web server and the web content is named and located as URL (Uniform Resource Locator). The content of the web servers is given in specific content types like hyper-text markup language (html) and Javascript, which are supported by the browsers also. Finally, the communication is performed with standard communication protocols, hyper-text transport protocol (http). During the communication, www protocols use three types of servers. The origin server provides the service that the client needs. The proxy server is located between the client and the server, where clients seek service from the proxy it acts as if it is the origin server, but it only forwards the request to it. In this way, the network load and the delay can be reduced as the proxy not only forwards the www content but also stores it and in case of another request for the same data serves it without the involvement of the origin server. The gateway server accepts a request as if it were the origin server. The system uses gateways for data and protocol conversion purposes.

2.2.2. *The WAP model*

The WAP architectural model (Fig. 2) is quite similar to the Web model. WAP application developers can use the same client–server architecture. WAP clients use the same URL name space, but certain protocols and languages are different. The WAP browser, the so-called micro-browser, runs on hand-held mobile devices and hence is not able to display html pages directly. These pages are translated by a gateway server to WML, the WAP native language. The WAP protocol stack provides communication. To sum up, WAP includes the following base components.

- A standard naming model — the same URL name space like in www.
- Content typing — WAP clients use standard content types.
- Content formats — WAP browsers can request WWW server content directly or through a gateway. WAP has extended the standard WWW formats to those that can be useful for mobile applications.
- A standard protocol stack that was developed for wireless networks.
- WAP proxy which involves a protocol gateway and content encoders and decoders. Encoders and decoders translate WAP contents into compact encoded format, which is optimal for mobile networks.

Figure 3 depicts an example of WAP network. The hand-held mobile client can communicate with both the WTA (Wireless Telephony Application) server which is the origin or gateway server of WAP and the web server. The WTA server content is WML (Wireless Markup Language) but the web server may also contain HTML pages. Thus, an HTML filter should be used to translate web content into WAP content to provide access to these pages for WAP clients. WAP proxy translates WAP requests to WWW, whereby it allows a WAP client to submit requests to the web server.

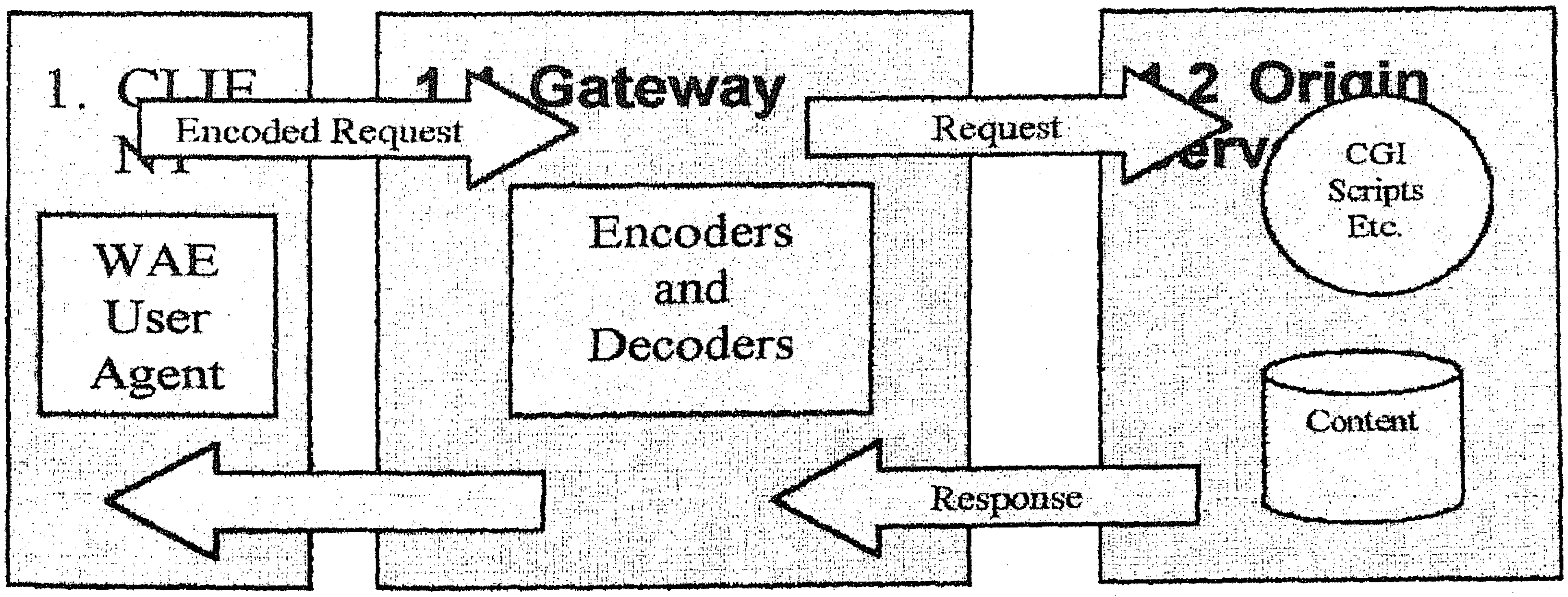


FIG. 2. WAP programming model.

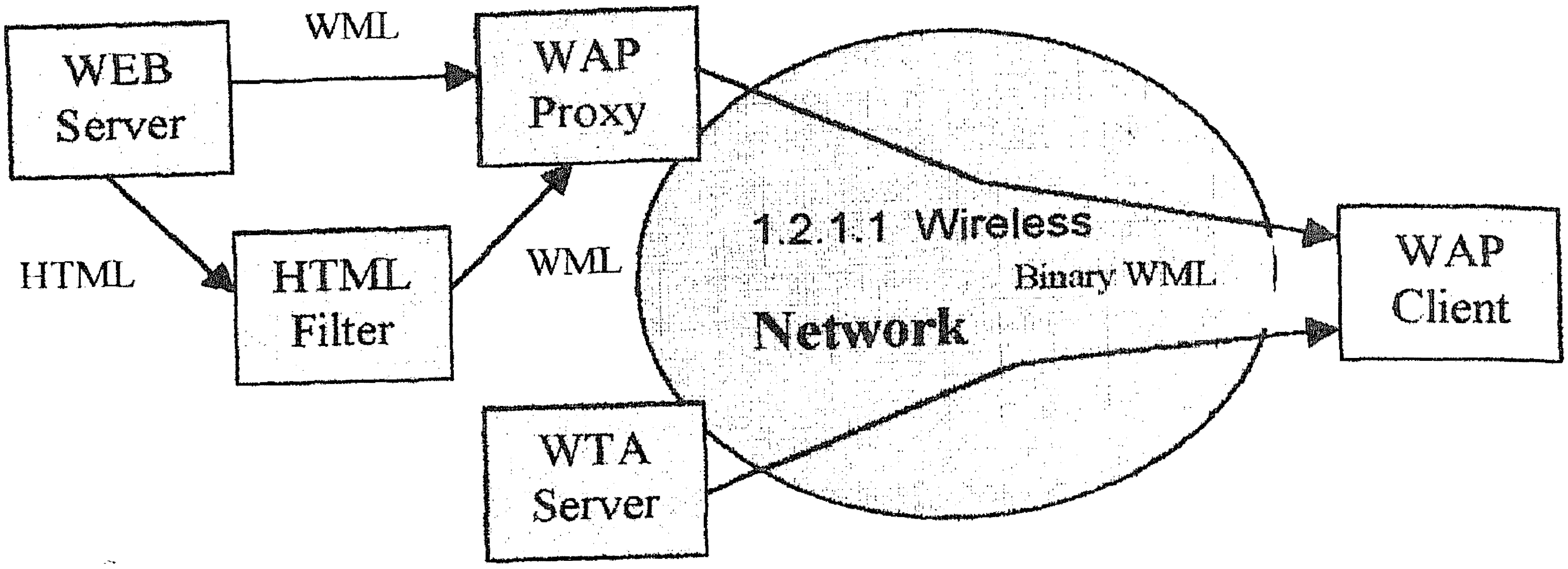


FIG. 3. Example of WAP network.

2.3. *WAP protocol hierarchy*

Figure 4 shows the hierarchical system of WAP. Due to the special constraints of wireless environment, the architectural structure has to be designed as flexible as possible.

2.4. *Wireless Application Environment (WAE)*

Even though the WAP applications can access all layers,[2] WAP Forum makes efforts to develop an application interface that satisfies as much demand as possible. WAE[3] makes easier to

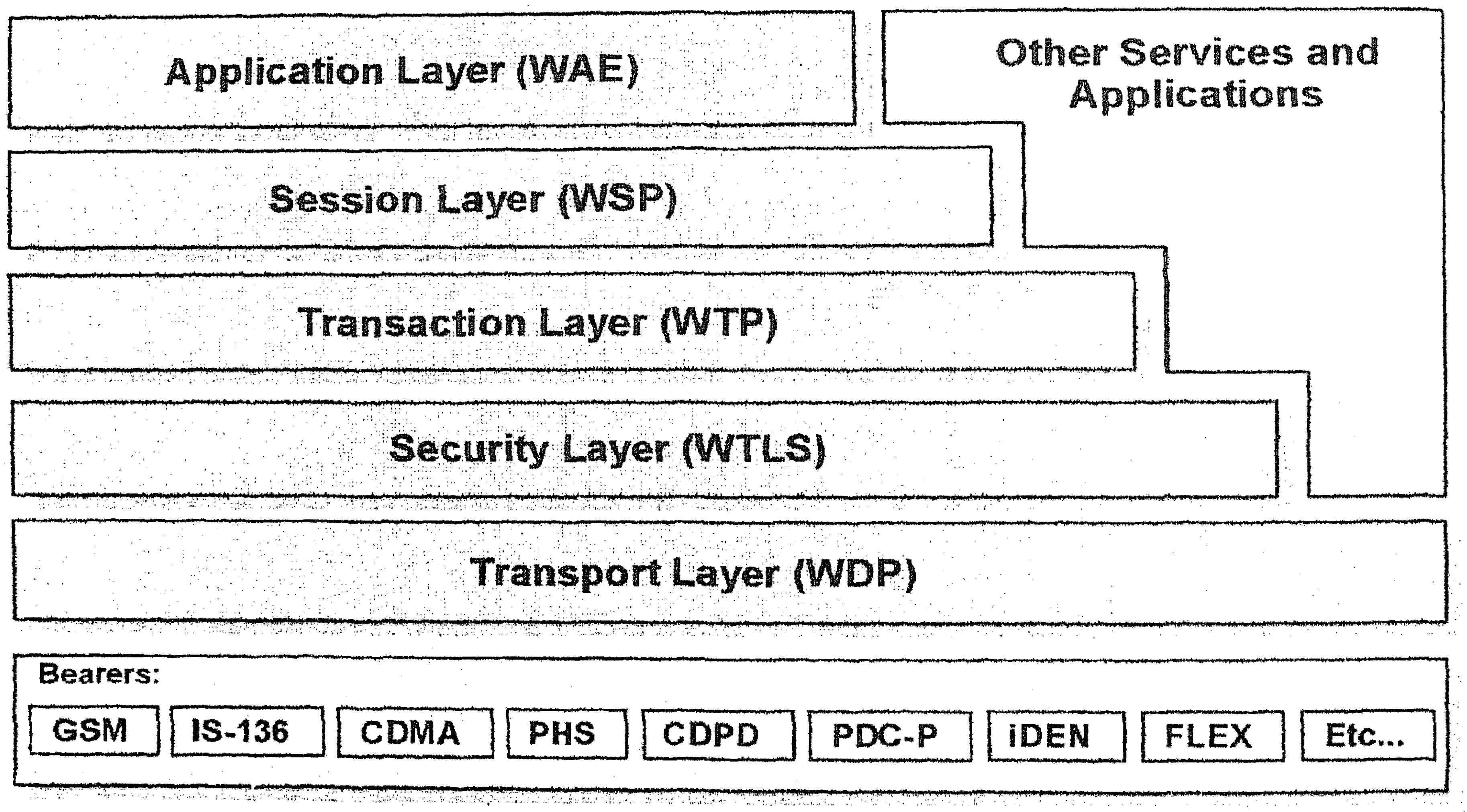


FIG. 4. WAP hierarchical system.

build such WAP applications, which can operate on several different wireless environments. WAE contains the following objects:

- Wireless Markup Language (WML) — a markup language similar to HTML, but wml is more suitable for mobile environment.
- WMLScript — a scripting language with similar functionality as JavaScript.
- Wireless Telephony Application (WTA, WTAI) — programming interfaces and telephony services
- Content formats — a couple of standard data formats for images, phone book records or calendar information

2.5. *Wireless Session Protocol (WSP)*

WSP protocol[4] is optimized for low-bandwidth bearer networks with relatively long latency. Wireless Session Protocol is the application layer of WAP that provides two kinds of services. The first one is for connection-oriented communication based on the underlying WTP layer and the second is for connectionless communication based on the unreliable and non-secure WDP layer. WSP currently consists of services for browsing applications, with the following functionality:

- HTTP/1.1 functionality and semantics in a compact over-the-air encoding.
- Long-lived session state.
- Session suspending and resuming with session migration.
- Common facilities for reliable and unreliable data push.
- Protocol features negotiation.

2.6. *Wireless Transaction Protocol (WTP)*

WTP[5] is a transaction-based protocol and runs on the top of a datagram service. Detailed description of WTP is given in the next section.

2.7. *Wireless Transport Layer Security (WTLS)*

In the wireless environment, secure communication is a very important issue. WTLS[6] is located on the top of WDP and under WTP and provides security for WAP applications. WTLS has modular architecture, thus the level of security depends on the communicating applications. WTLS functionality is a secure solution to the following topics:

- Data integrity: WTLS contains facilities to ensure that data sent between applications is unchanged and uncorrupted.
- Privacy: Confidential data cannot be understood by unauthorized applications or users.
- Authentication: Depending on the given application the client has to authenticate before retrieving data.
- Denial-of-service protection: WTLS contains facilities for detecting and rejecting data that are replayed or are not successfully verified. WTLS makes typical denial-of-service attacks harder to accomplish and protects the upper protocol layers.

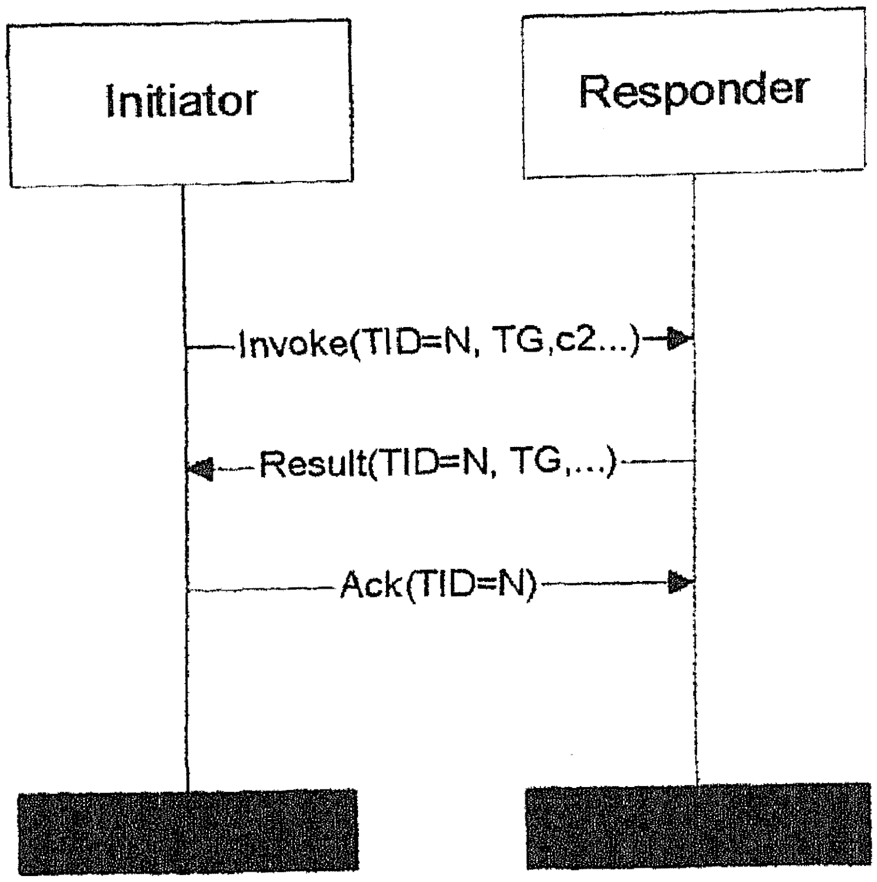


FIG. 5. Basic class-2 transaction.

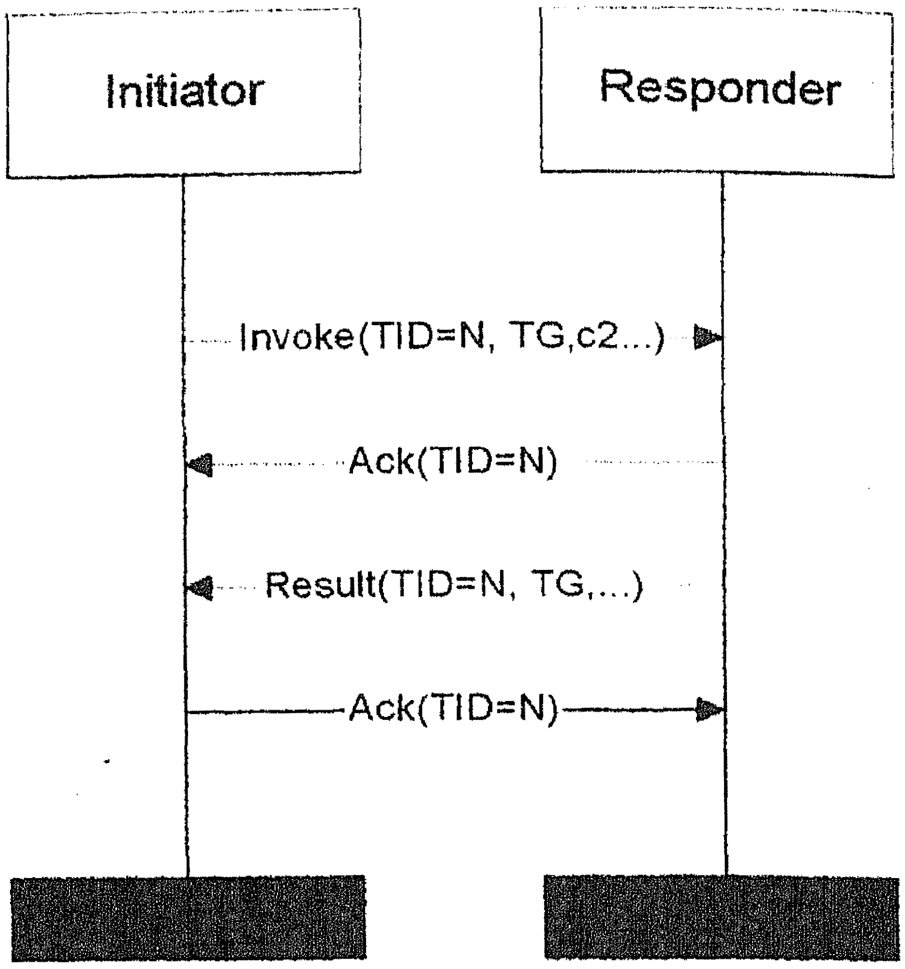


FIG. 6. Class-2 transaction with 'hold-on' acknowledgement.

2.8. *Wireless Datagram Protocol (WDP)*

WDP[7] provides a general transport service for WAP. WDP can operate above several bearer services, so WAP applications and the higher layers do not depend on the specific wireless networks. On the other hand, WAP applications can use other transport services like TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) which rely on IP (Internet Protocol). Thus, WAP applications are independent of wireless environment.

2.9. *Bearers*

WAP has been designed to combine the advantages of Internet and mobile technologies. Thus, WAP has to support as many types of bearer as possible including short message, circuit-switched data and packet data. Unfortunately, the different bearer types have different features, like quality of service. WAP protocols have been designed to compensate for or tolerate varying levels of service. WDP establishes the connection between the higher level protocols and bearer services, so the several bearer types supported by WAP can be found in WDP specification.[7]

3. A WAP Protocol: WTP

3.1. *Overview of WTP*

Wireless Transaction Protocol (WTP) operates on the top of the datagram and security layer and provides both reliable and unreliable service for the WTP user. Reliability is achieved through the use of unique transaction identifiers, acknowledgements, duplicate removal and retransmissions. Optionally, WTP provides user-to-user reliability, thus WTP user confirms every packet. WTP has to cope with the special conditions of wireless environment, so it uses implicit acknowledgement as much as possible and tries to minimize the number of message retransmissions due to duplicate packets. WTP makes possible to concatenate several messages into one Protocol Data Unit (PDU) and the last acknowledgement may contain extra information related to communication, for example, performance measurements.

3.1.1. *Transaction classes*

WTP is a transaction-based protocol. It provides three types of transaction classes that realize the unreliable and reliable message transfer. The launching side of communication is referred to as the initiator. The responding side is referred to as the responder. The transaction class is defined by the initiator and cannot be negotiated.

Class 0 transaction is created for unreliable communication. Basically it can be used for an unreliable push service. The procedure of such a transaction is as follows: One invoke message is sent by the initiator to the responder. The responder does not reply at all and does not send an acknowledgement. The transaction ends when the Invoke has been received. The transaction is stateless and cannot be aborted.

Class 1 transaction provides a reliable service. It can be used for reliable push service. The basic operation of this transaction is as follows: The initiator sends an Invoke message to the responder but, unlike the previous type, the responder sends an acknowledgement. If the acknowledgement packet does not arrive in time, the initiator retransmits the Invoke message. In this case the responder sends the acknowledgement packet again.

Class 2 transaction involves a reliable Invoke message and a reliable result message. One of the great advantages of WAP is that mobile devices can be used to reach Internet and WAP databases. The primary function of Class 2 transaction is to serve such browsing applications of WAP. The basic behaviour of this transaction type is as follows: The initiator sends an Invoke message to the responder. If the responder is not able to process the Invoke message in time, it will send a hold-on-acknowledge to avoid the Invoke retransmission. The responder sends back a result message, which is the implicit acknowledgement of Invoke. Finally, the initiator acknowledges this result message. If the acknowledgement message gets lost, the responder will retransmit the result message. Accordingly the initiator has to retransmit the acknowledgement.

3.2. *Protocol features of WTP*

3.2.1. *Retransmission until acknowledgement*

Retransmission is used to guarantee reliable transfer of data from one WTP provider to another. WTP uses implicit retransmissions as much as possible to decrease the load of the network. When any of the WTP providers sends a message, it sets the retransmission counter to zero and starts the retransmission timer. If the retransmission timer expires, the retransmission counter will be increased by one and the timer will be set to zero again. If the retransmission counter reaches the maximal value, the transaction will be aborted.

When retransmission occurs, the retransmission flag is set in the WTP message. This flag indicates to the addressee that this message is due to retransmission and it is not a duplicated message.

3.2.2. *User acknowledgement*

This feature of WTP makes it possible for the WTP user to confirm every message. When this function is enabled, WTP cannot send an acknowledgement, but just sends an indication to the WTP user and waits until the upper layer releases the response primitive. This function is optional; however, WSP needs it, so such WAP devices that include the whole protocol stack have to implement it.

If the initiator wants to use user acknowledgement, it will set the appropriate flag in the Invoke message. When the message arrives to the responder, it forwards it to the WTP user, does not send an acknowledgement and starts a timer. However, this timer can expire later than initiator retransmission timer; so, if the initiator retransmits the message the responder has to discard it and reset the timer. If the upper layer does not respond until a certain time, the responder will abort the transaction.

3.2.3. *Information in last acknowledgement*

The WTP user can attach extra information to the last and only the last acknowledgement. This information can describe the quality of transmission, network congestion or load. The extra information is stored in a Transport Information Item (TPI). The function and types of TPIs are described later in detail.

3.2.4. *Concatenation and separation*

WTP tries to save the network bandwidth as much as possible. Concatenation puts several WTP messages to one packet of the datagram service. Separation is a procedure to extract these messages after the arrival and separate again for the WSP. In other words, concatenation puts more WAP PDU (Protocol Data Unit) to one datagram service SDU (Service Data Unit) and separation extracts them again.

Concatenation and separation can be used if the messages have the same port and address information, so they belong to the same WAP application. Implementation of this function is not specified. WTP provides only the structure to be used when multiple messages are concatenated.

3.2.5. *Transaction Identifier (TID)*

TID is a 16-bit long integer value and uniquely identifies, with source and destination port and address, the transaction. The initiator increments the TID by one for each initiated transaction. This means that TIDs 1, 2 and 3 can go to server A, TIDs 4, 5 to server B and 6 to server A again. However, when a message is retransmitted, the TID is reused. Thus, the responder can filter out the new Invoke messages from duplicated and old ones. The new Invoke always has a higher TID value. Nevertheless, if two WAP devices communicate with each other, both of them can simultaneously be initiator and responder, so they can assign the same TID to different transactions. Thus, the higher bit of TID is always set to zero by the initiator and one by the responder. The TID can be validated at any time in the course of the communication, to ensure a reliable connection.

3.2.6. *Transport Information Items (TPI-s)*

The WTP header contains a variable part. This portion of the header may contain the TIPs. If the header does not include TPIs, the variable part must be empty. The possibility that TPIs has several reasons. On the one hand, protocol flexibility has been increased by them, because future developments can be implemented by introduction of new TPI types. On the other, the WTP header always contains only the relevant information.

Currently, four TPI types are specified. 'Error TPI' is used, when an unsupported TPI is received. 'Info TPI' is defined for information in last acknowledgement function. 'Option TPI' can be used for negotiating default values of timers and counters. 'Packet Sequence Number TPI' contains sequence number of segmented WAP PDUs and introduced into the segmentation and re-assembly procedure.

3.2.7. *Segmentation And Re-assembly (SAR)*

WTP can operate on unreliable datagram services like IP (Internet Protocol). Thus, for reliability it has to support segmentation and re-assembly function. However, this feature is optional but if it is not implemented the datagram layer has to support it. Segmentation means that WTP sends a message in several packets while packet size is adapted to the packet sizes of the bearer network. The receiver can demand selective retransmission in case of packet loss, so it is not needed to retransmit the whole message. Re-assembly means putting the whole message together. This procedure is based on the packet sequence number. Packets belonging to one message are grouped and sent and acknowledged together.

3.3. *Examples of WTP operations*

3.3.1. *Class 2 transaction*

1. The initiator initiates a class-2 transaction.
2. The responder waits for the invoke message to be processed and implicitly acknowledges with the result.
3. The initiator acknowledges the received result message.

3.3.2. *Transaction with 'hold-on' acknowledgement*

1. The initiator initiates a class-2 transaction.
2. The responder waits for the invoke message to be processed. The acknowledgement timer at the responder expires and 'hold-on' acknowledgement is sent to prevent the initiator from re-transmitting the invoke message.
3. The result is sent to the initiator.
4. The initiator acknowledges the received result message.

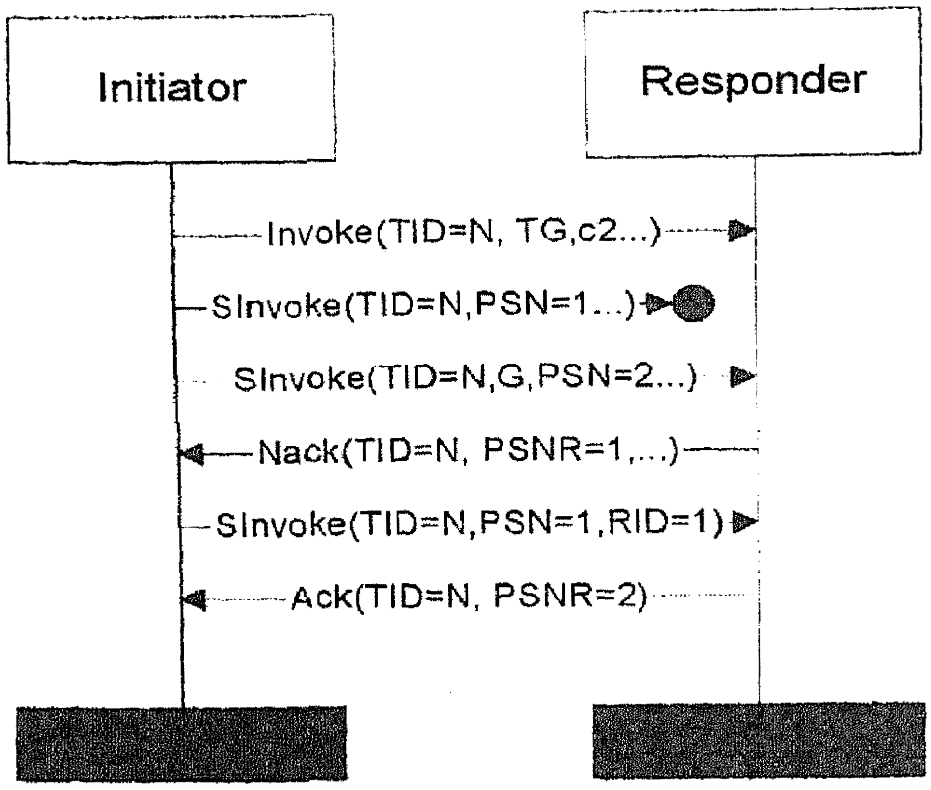


FIG. 7. Selective retransmission.

3.3.3. *Selective retransmission*

Figure 7 depicts the case when the Invoke message is segmented, but the first packet has been lost. The responder cannot reassemble the packet due to the missing segment. Then a negative acknowledgement is sent back to the initiator and the missing packet is retransmitted. In the retransmitted PDU the retransmission indicator (RID) flag is set. Once the responder has received the retransmitted packet, the message is acknowledged and the transaction is finished.

4. Specification

An informal specification can be found in WAP Forum WTP specification where the WTP protocol dynamic behaviour and the used messages are described. To implement the specified protocol it is not enough to have an informal specification. For formal specification we are using high-level languages MSC[8], SDL[9] and for data ASN.1.[10] This formal specification has a double reason. Firstly, it helps for the implementation of the protocol where this formal description is transformed to a 'computer' language. On the other hand, the formal specification of the protocol will help us to generate test cases to this protocol. The dynamic behaviour of the protocol can be modeled with SDL. It is very important to decide previously the structure of the SDL system because we must care not only about the communication between the neighbour layers but we also have to see the whole protocol stack from the upper side. There are several possibilities to solve this problem. If we regard this problem from the point of view of generating test cases we only have to describe the interfaces on WTP layer. In this case we can follow the system SDL as represented in Fig. 8.

Of course, this system description can be extended easily to other layers too, and finally, the stack external behaviour can be modeled. The communication via UT will be placed via UTr channel and in this way the service primitive communication will be realized. In the other channel, the PDUs will communicate through the WDP and service provider. In the WTP block we can find the initiator and the responder processes which will describe the protocol

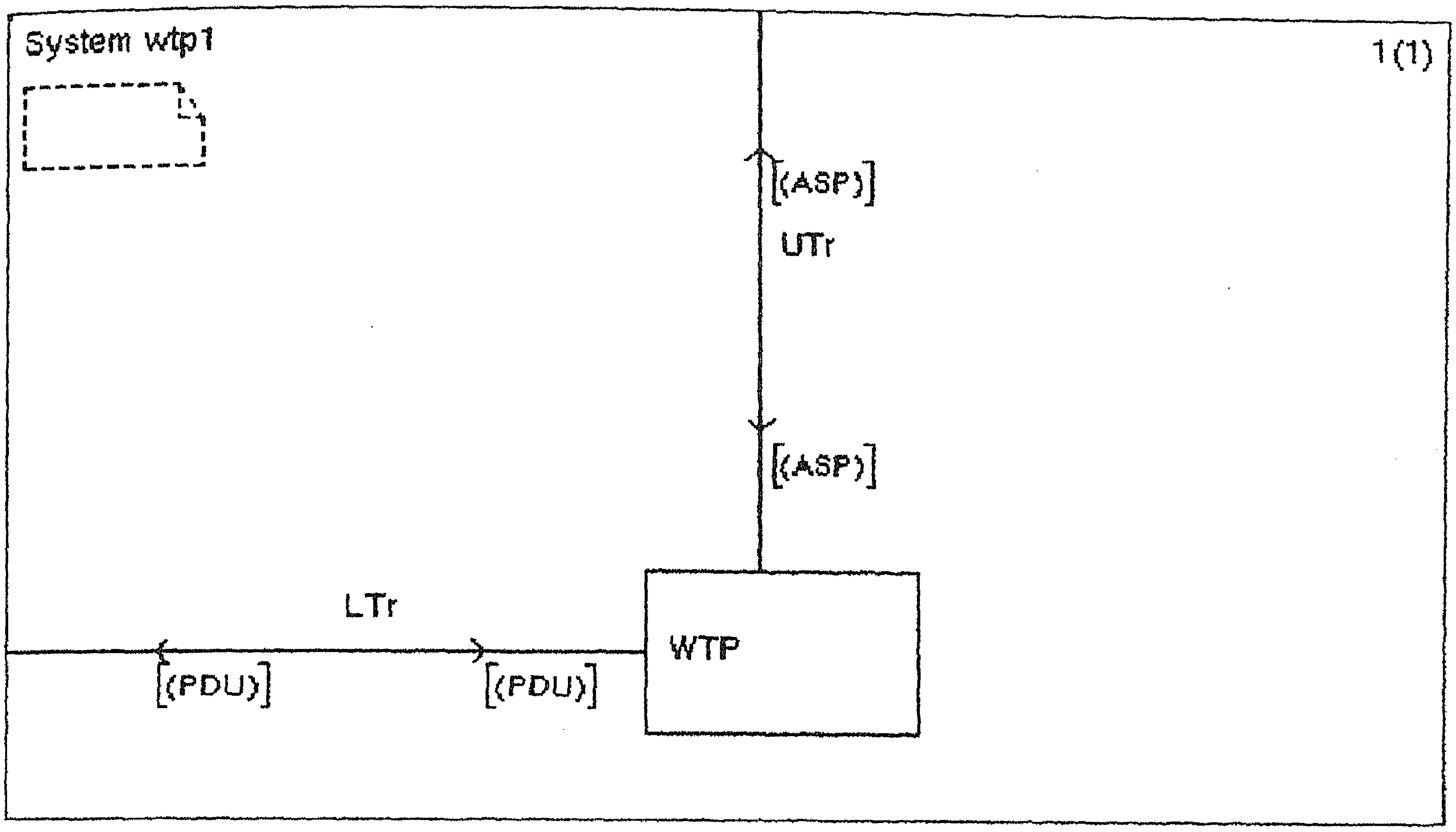


FIG. 8. SDL system description.

machine. The two separate processes can be found in the block diagram in Fig. 9. In this case, similar to the precedent case in the signal routes UT_si and UT_sr, the ASPs communicate the UT and the PDUs in the case of LT_si and LT_sr. After designing the responder and initiator processes we can generate test cases using Telelogic SDT.

5. Software testing

Protocols are key elements of a telecommunication system, and hence their reliability is important. In the present communication system, the software has great importance, and the correctness of the system will determine how reliable it is. The software protocol has to work with each other and with the products of other firms and meet the specifications of a standard. Software testing checks the correctness of the system and is very important as we cannot interact with a protocol before the software communicates with its environment. WAP software testing will check if the implementation covers the minimum requirements specified in the WAP standard. These are not only functional requirements, but also the quality is checked and performance measurements verified.

5.1. *The notion of software testing*

The generic steps of software testing are the following:

- developing the general test project plan
- developing the test case suite specifying the functional groups

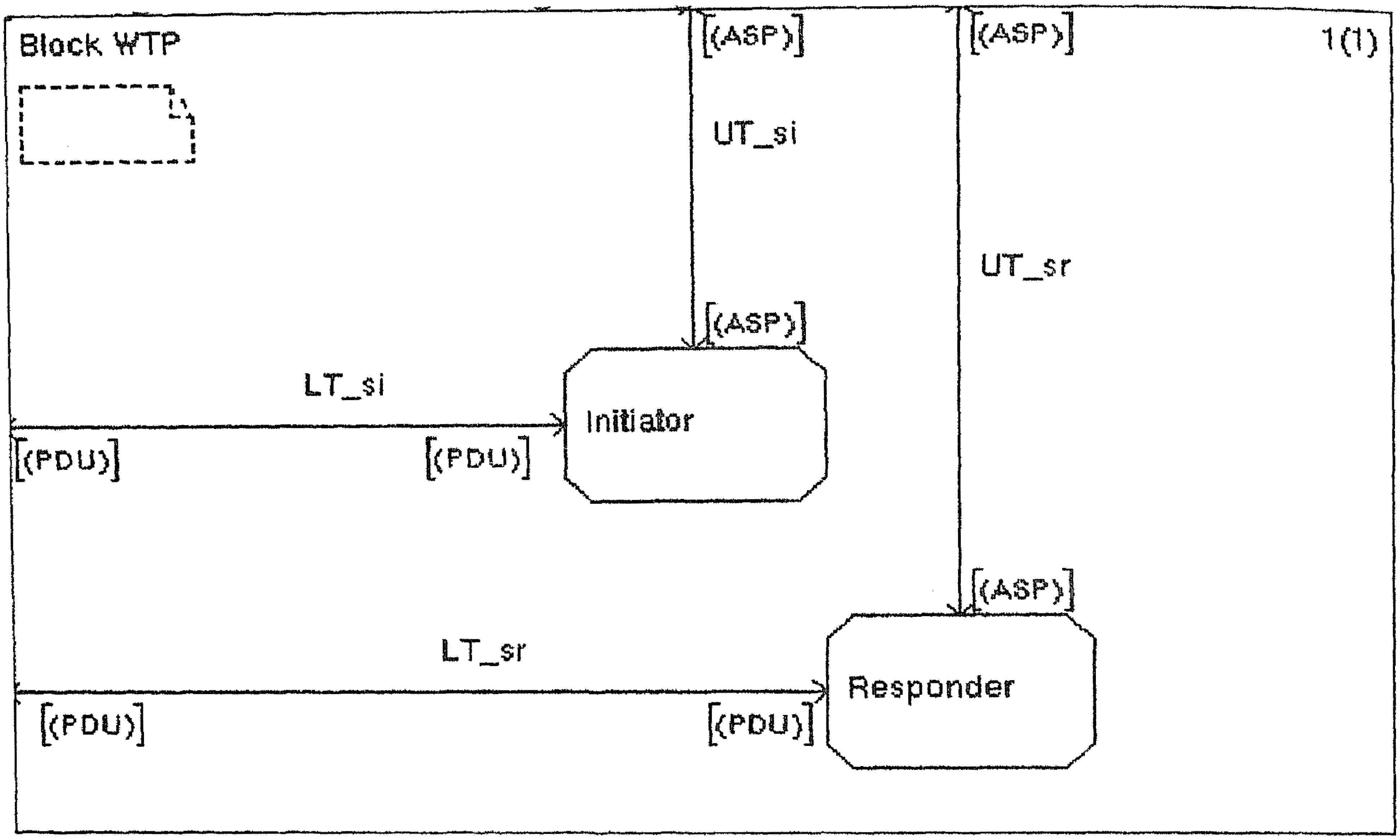


FIG. 9. WTP block structure.

- implementing the test cases
- running test cases
- generating or manufacturing a test report

The test project plan will contain all test purposes, the used test architecture, and all specific information, which is useful in the testing procedure.[11]

5.2. *Test standardization principles*

Standardization of the software test process is very important because the whole testing procedure can be partitioned into many parts and thus can be coordinated much easier. The following principles are important to employ the knowledge independent of human interaction:

- Hierarchical grouping of the test cases—it is easier to find something if we know where to find it, and it is simpler to construct more parts.
- Identifying the test cases—it is an interface between the test case and the operator to know what each test case does.
- Test report database—it is very important to know the result of the test cases, for it is useful to construct a database to know the test results or the test logs.
- Good and continuous relation with the developers—there must be a good teamwork and a good communication between the testers and developers.

- Documenting—it is a basic concept that the work must be documented for reuse of the necessary parts later.

We have seen that the WTP layer contains more than 100 test cases. Executing them manually is a hard task and will require manual intervention to complete the whole test process. It is necessary to develop an automatic test execution process, which will execute each test case step by step from a well-specified list and test all cases, if required. This automated process must provide some obligatory details like test architecture, the test executor's name, the date and the time of the testing procedure, etc to the test environment and the test executor.

The process described herein is only one of the possible solutions. The benefits of this process are:

- It can be used generally for any other software testing.
- Knowledge of any specific language is not necessary to execute the tests.
- The execution time is minimized due to automation.
- Continuous preparation of test cases can be synchronized to the developing process and done in parallel.
- Test suite can be implemented in an easy and inexpensive manner.
- Easy to realize.

This software-testing scenario is necessary to use in the early development phase of the protocol when the implementation is not perfectly executed or when we need relatively rapid response to the development team.

6. Conformance testing

6.1. *WAP conformance testing*

Conformance testing[12] is a process of verifying the formal requirements of the reference standards and more precisely that it meets the conformance clauses contained in the standards. During the test phase, the implementation is referred to as the Implementation Under Test (IUT). The primary objective of conformance testing is to increase the probability that different product implementations can actually interoperate. The method used to exercise the test, called the conformance assessment process, is standardized to achieve some degree of comparability of test results of similar products tested by different laboratories.

Conformance can be divided into two broad sub-categories, static and dynamic conformance. Static conformance specifies the limitations on the combinations of implemented capabilities permitted in a real open system. Dynamic conformance specifies what observable behavior is permitted by the relevant standard in instances of communication.

With respect to WAP, the static conformance clause for each specification within the WAP stack would define the minimum set of protocol features or options that can be implemented to ensure that each implementation will be able to interoperate with other implementations. This would include defining combinations of optional features that must be implemented together in order to achieve successful interoperability. The WAP dynamic conformance clause would

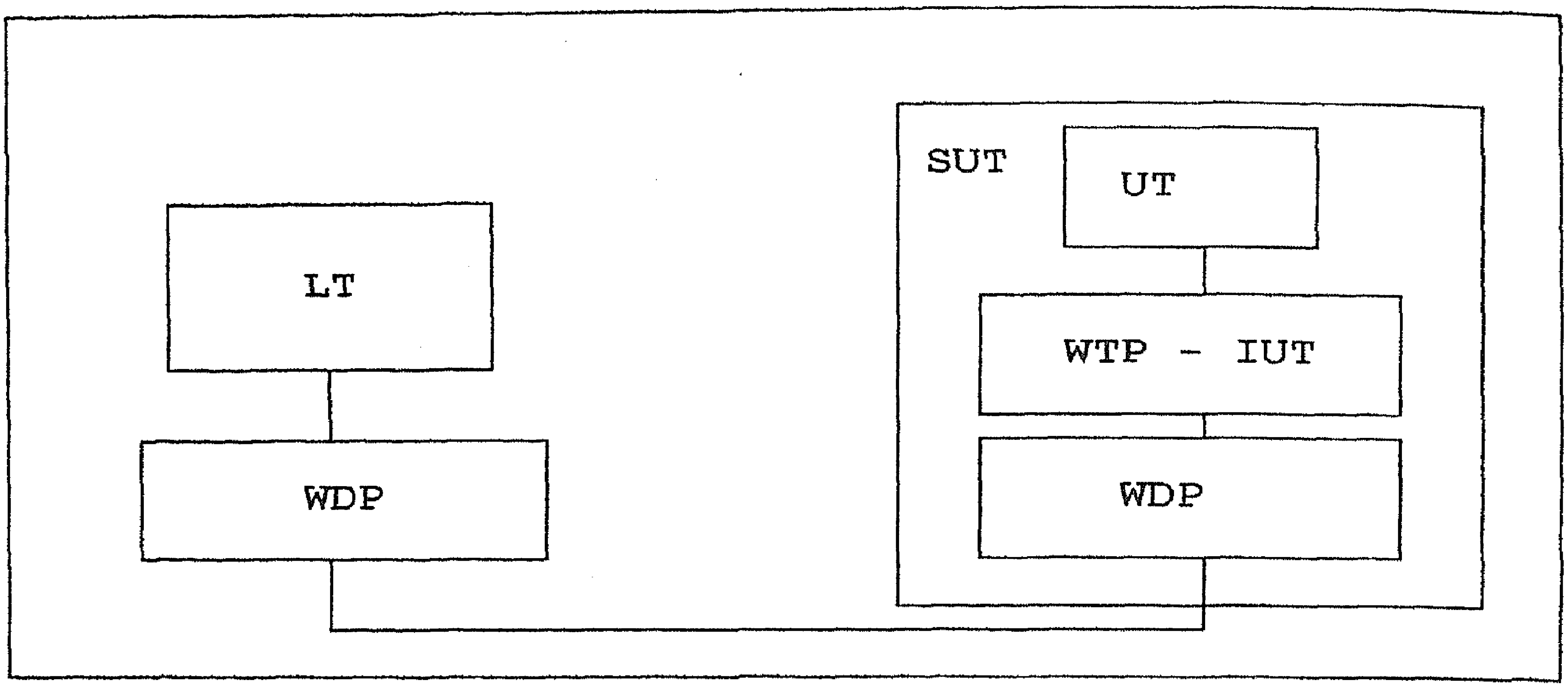


FIG. 10. The reference test configuration for the WTP protocol.

define how the protocol is to behave when receiving a PDU from its peer or receiving a service primitive from a higher layer. These involve the development of executable test cases and test suites, which can be run against an implementation.

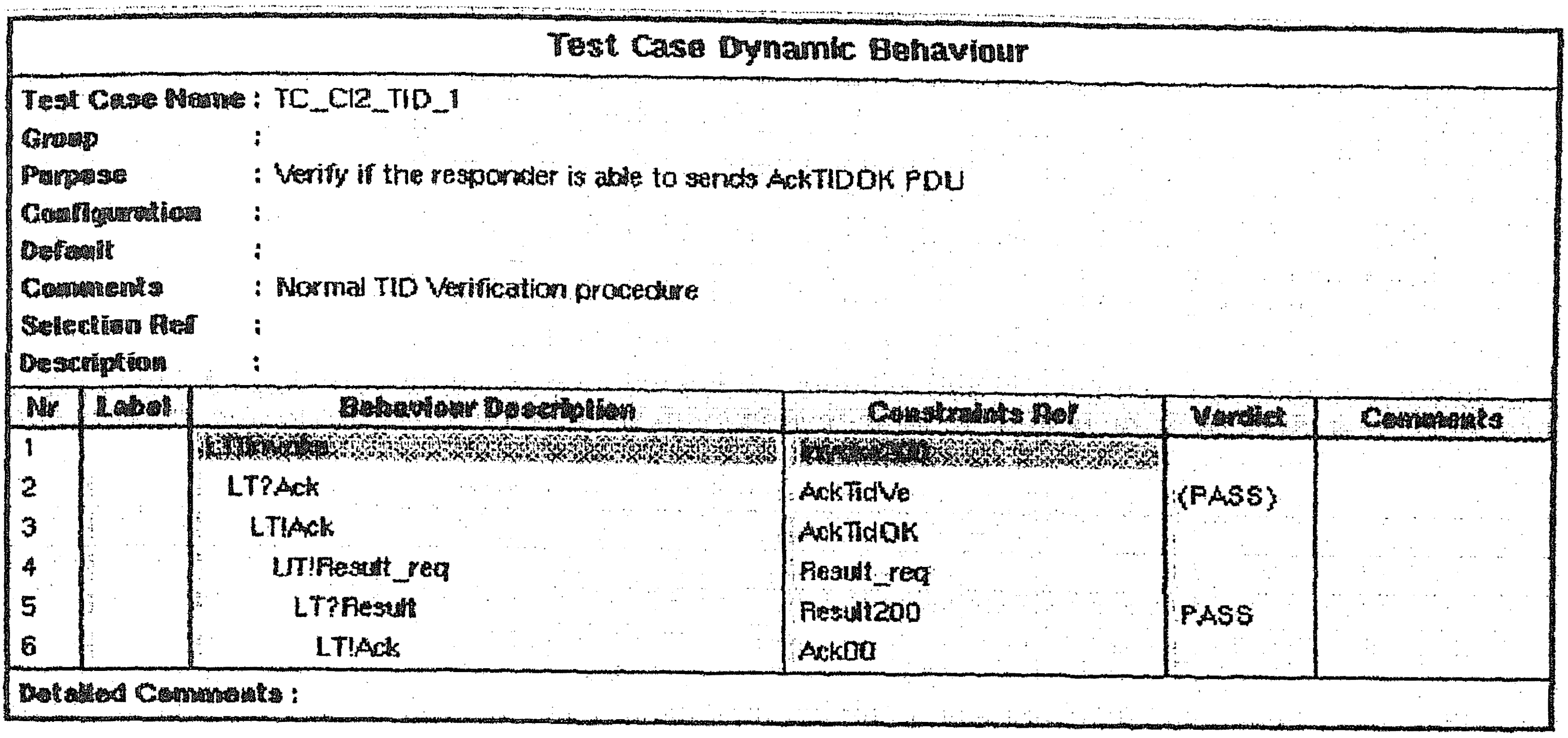
The WAP Forum will develop and standardize these tests in the near future.

6.2. *Conformance testing of WTP protocol*

The main difference between conformance and software testings is that in the case of conformance testing we use the notion of black box testing, the protocol behaviour can be observed and controlled only via its interfaces. The reference test configuration can be seen in Fig. 10, where the LT represents the lower tester and is situated on the test system.

The IUT is situated on the SUT (System Under Test) above the WDP layer. Over the IUT, the UT (Upper Tester) is located and sends TR service primitives to the WTP layer. The PDUs are transmitted through the WDP layer from the LT. The test suite of WTP protocol is described in TTCN (Tree and Tabular Combined Notation).[13]

The first step to develop the test cases is to determine the test suite structure where the test cases are classified according to functional requirements. After the TSS development we have to determine the data part of the protocol (PDU, ASP, Timer values, etc.) and the interface function between the data types and the 'real' system. In the first step the test cases are classified according to the types of services. On the test group, which will control the class two transaction, there are more test groups for different protocol features. In this case the communication will begin with an Invoke, which has TCL = 2 as parameter. Of course, the LT is the initiator in this case. The IUT is the responder and will transmit an Ack PDU with TIDVe flag set, which means that it will ask a TID verification procedure required (Fig. 11).



Test Case Dynamic Behaviour

Test Case Name : TC_Cl2_TID_1
Group :
Purpose : Verify if the responder is able to sends AckTIDOK PDU
Configuration :
Default :
Comments : Normal TID Verification procedure
Selection Ref :
Description :

Nr	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		LT!Invoke	Invoke200		
2		LT?Ack	AckTidVe	(PASS)	
3		LT!Ack	AckTidOK		
4		UT!Result_req	Result_req		
5		LT?Result	Result200	PASS	
6		LT!Ack	Ack00		

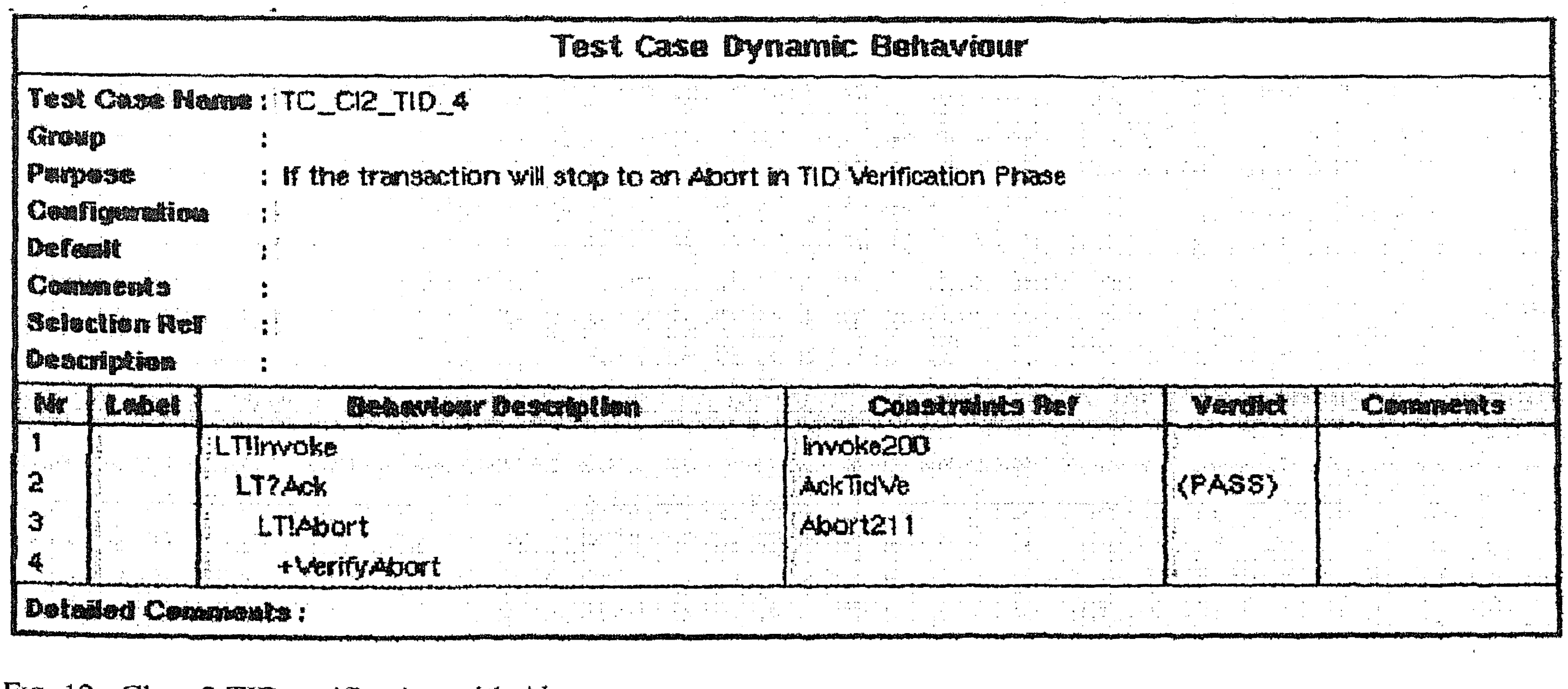
Detailed Comments :

FIG. 11. Class 2 TID verification.

After Ack TIDVe, the LT will send an Ack TID OK set and the UT a TR-Result.req service primitive. In this case, the LT will receive a Result PDU as described in Fig. 11. To complete the transaction, the LT will send an Ack PDU.

In Fig. 12 we can see another case where the LT will send an Abort PDU (USER Abort) to the responder and will abort the transaction.

The goal is to develop a set of test cases, which will cover the maximum part of the protocol behaviour.



Test Case Dynamic Behaviour

Test Case Name : TC_Cl2_TID_4
Group :
Purpose : If the transaction will stop to an Abort in TID Verification Phase
Configuration :
Default :
Comments :
Selection Ref :
Description :

Nr	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		LT!Invoke	Invoke200		
2		LT?Ack	AckTidVe	(PASS)	
3		LT!Abort	Abort211		
4		+VerifyAbort			

Detailed Comments :

FIG. 12. Class 2 TID verification with Abort.

7. Conclusions

The WAP-based solutions are really useful in many application fields, containing Internet communications and advanced telephony services on digital mobile phones, pagers and other wireless terminals. This paper gave a short overview of the technical aspects of how WAP enables corporations, system integrators and software developers to offer Internet- and Intranet-based information and services to their customers, business partners and employees through a mobile phone.

Finally some personal remarks. One of the authors (KT) visited the Indian Institute of Science in Bangalore four years ago and she got the best impression from the research activity as well as the advanced degrees in the postgraduate level, especially in protocol engineering. One of the world's leading mobile phone suppliers, Nokia, has signed new research cooperation agreements with the Institute. One of the possible research topics is WAP interoperability testing. This final personal remark reflects well the strength of worldwide WAP research and development opening new ways and supporting value-added services in the near future.

8. Abbreviations

ASN. 1	Abstract Syntax Notation One
ASP	Abstract Service Primitive
CDMA	Code Division Multiple Access
CDPD	Cellular Digital Packet Data
CGI	Common Gateway Interface
GSM	Global System for Mobile Communication
HTML	Hyper-Text Markup Language
HTTP	Hyper-Text Transport Service
IDEN	Integrated Digital Enhanced Network
IP	Internet Protocol
IUT	Implementation Under Test
LT	Lower Tester
MSC	Message Sequence Charts
PDA	Personal Digital Assistant
PDU	Protocol Data Unit
PHS	Personal Handy Phone
RID	Retransmission Indicator
SAR	Segmentation and Re-assembly
SDL	Specification and Description Language
SDT	SDL Design Tool
SDU	Service Data Unit
SUT	System Under Test
TCP	Transmission Control Protocol
TID	Transaction Identifier
TPI	Transport Information Item
TSS	Test Suite Structure
TTCN	Tree and Tabular Combined Notation

UDP	User Datagram Protocol
URL	Uniform Resource Location
UT	Upper Tester
WAE	Wireless Application Environment
WAP	Wireless Application Protocol
WDP	Wireless Datagram Protocol
WML	Wireless Markup Language
WSP	Wireless Session Protocol
WTA	Wireless Telephony Application
WTAI	WTA Interface
WTLS	Wireless Transport Layer Security
WTP	Wireless Transaction Protocol
WWW	World Wide Web

References

1. WAP Architecture Specification, WAP Forum, http:/www.wapforum.org/

2. Wireless Application Environment Overview, WAP Forum, http://www.wapforum.org/

3. Wireless Application Environment Specification, WAP Forum, http://www.wapforum.org/

4. Wireless Session Protocol, WAP Forum, http://www.wapforum.org/

5. Wireless Transaction Protocol, WAP Forum, http://www.wapforum.org/

6. Wireless Transport Layer Security Protocol, WAP Forum, http://www.wapforum.org/

7. Wireless Datagram Protocol Specification, WAP Forum.

8. ITU-T Z.120(1993), Message Sequence Charts.

9. ITU-T Z.100(1994), CCITT Specification and Description Language (SDL)

10. ITU-T, Abstract Syntax Notation One (ASN.1).

11. LÁSZLÓ SZÜCS, *Standardization and automatization of software tests*, Hungarian Telecommunication, July 1999.

12. BAUMGARTEN, B. AND GIESSLER, A. *OSI Conformance Testing Methodology and TTCN*, Elsevier, 1994.

13. ITU-T X292 OSI Conformance Testing Methodology and Framework for protocol recommendations for CCITT Applications, The Tree and Tabular Combined Notation.

14. TARNAY, K. *Protocol specification and testing*, Plenum Press, 1991.