International Telecommunication Union
Standardization Sector

Supplement

ITU-T P Suppl. 30 (09/2023)

SERIES P: Telephone transmission quality, telephone installations, local line networks

Supplements to ITU-T P-series Recommendations

Considerations on the automation of digital financial services testing

ITU-T P-SERIES RECOMMENDATIONS

Telephone transmission quality, telephone installations, local line networks

Vocabulary and effects of transmission parameters on customer opinion of transmission quality	P.10-P.19
Voice terminal characteristics	P.30-P.39
Reference systems	P.40-P.49
Objective measuring apparatus	P.50-P.59
Objective electro-acoustical measurements	P.60-P.69
Measurements related to speech loudness	P.70-P.79
Methods for objective and subjective assessment of speech quality	P.80-P.89
Voice terminal characteristics	P.300-P.399
Objective measuring apparatus	P.500-P.599
Measurements related to speech loudness	P.700-P.709
Methods for objective and subjective assessment of speech and video quality	P.800-P.899
Audiovisual quality in multimedia services	P.900-P.999
Transmission performance and QoS aspects of IP end-points	P.1000-P.1099
Communications involving vehicles	P.1100-P.1199
Models and tools for quality assessment of streamed media	P.1200-P.1299
Telemeeting assessment	P.1300-P.1399
Statistical analysis, evaluation and reporting guidelines of quality measurements	P.1400-P.1499
Methods for objective and subjective assessment of quality of services other than speech and video	P.1500-P.1599

For further details, please refer to the list of ITU-T Recommendations.

Supplement 30 to ITU-T P-series Recommendations

Considerations on the automation of digital financial services testing

Summary

When testing digital financial services (DFS), a variety of specific aspects have to be considered. First, these services are an unprecedented combination of efforts in both the telecommunication and the financial sector. Secon, and most important, each test carried out sends real money (which might be retrieved in many cases), and is being charged real fees (which typically cannot be retrieved and add to the cost of testing).

Once attempts are made towards automation in DFS testing, many more aspects and requirements need to be considered.

Supplement 30 to ITU-T P-series Recommendations provides guidance on aspects of automation in DFS testing that need to be included in the test conditions as well as specific instructions that experimenters should consider.

This Supplement also provides examples of automation that are suitable for DFS testing. It is meant to be a starting point and basis for industry vendors and other stakeholders in the DFS ecosystem to develop robust environments for automated testing.

In all cases, it is assumed that automation is technology neutral and technology agnostic.

History *

Edition	Recommendation	Approval	Study Group	Unique ID
1.0	ITU-T P Suppl. 30	2023-09-28	12	11.1002/1000/15711

Keywords

Digital financial services, QoE, QoS, quality of experience, quality of service, testing automation.

---

* To access the Recommendation, type the URL https://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this publication may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the publication development process.

As of the date of approval of this publication, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this publication. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

		Page
1	Scope	1
2	References	1
3	Definitions	1
	3.1 Terms defined elsewhere	1
	3.2 Terms defined in this Supplement	1
4	Abbreviations and acronyms	1
5	Conventions	2
6	Basics of testing digital financial services	2
7	Automating the core DFS use case	3
8	Testing conditions	4
9	Hybrid approaches	4
10	Special conditions for development and operation of automated DFS testing	5
11	Overview: Types of automated testing systems and main characteristics	5
12	Summary and conclusions	6

Supplement 30 to ITU-T P-series Recommendations

Considerations on the automation of digital financial services testing

1 Scope

This Supplement provides guidance on aspects of automation in digital financial services (DFS) testing that need to be included in the test conditions as well as specific instructions that experimenters should consider. This guidance is given under the assumption that automation is technology neutral and technology agnostic.

2 References

[ITU-T E.800] Recommendation ITU-T E.800 (2008), *Definitions of terms related to quality of service.*

[ITU-T P.10] Recommendation ITU-T P.10/G.100 (2017), *Vocabulary for performance, quality of service and quality of experience.*

[ITU-T P.1502] Recommendation ITU-T P.1502 (2020), Methodology for QoE testing of *digital financial services.*

[ITU-T P.1503] Recommendation ITU-T P.1503 (2023), *Extended methodology for cross-country and inter-operator digital financial services testing.*

3 Definitions

3.1 Terms defined elsewhere

This Supplement uses the following terms defined elsewhere:

3.1.1 quality of experience (QoE) [ITU-T P.10]: The degree of delight or annoyance of the user of an application or service.

3.1.2 quality of service (QoS) [ITU-T E.800]: The totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service.

3.2.3 digital financial service [ITU-T P.1503]: Methods to electronically store and transfer funds; to make and receive payments; to borrow, save, insure, and invest; and to manage personal or business finances; and where these services are accessed via mobile communication devices.

3.2 Terms defined in this Supplement

None.

4 Abbreviations and acronyms

This Supplement uses the following abbreviations and acronyms:

2G Second Generation mobile networks

DFS Digital Financial Service

KPI Key Performance Indicator

SMS Short Message Service

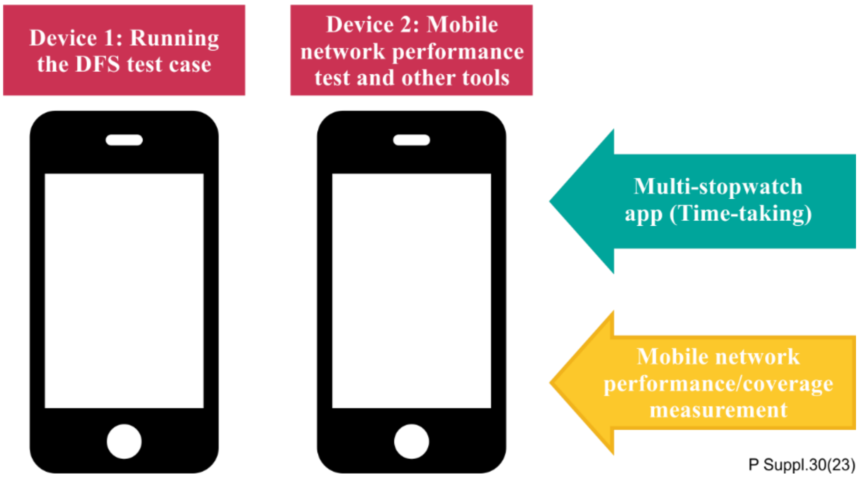
USSD Unstructured Supplementary Service Dat

5 Conventions

None.

6 Basics of testing digital financial services

As detailed in [ITU-T P.1503], the methodology of DFS testing uses a two-channel approach with two devices at either end of the money transfer chain.



Device 1 (left) is used to run the actual DFS use case. Device 2 (right) runs the mobile network performance testing (background test) and is also used to host auxiliary tools, in this case a multi-stopwatch app.

Figure 1 – Principle of two-channel approach for DFS testing

The first device is used to run the actual end-to-end test case, in this case, the P2P money transfer. The second device is used to run the mobile network performance test (background test), and additional tools to support testing. In the example shown in Figure 1, this is the "multi-stopwatch" app used for logging of DFS events.

In this basic setup, Device 1 can be any type of mobile device, including "feature phones", i.e., pre-smartphone types. For testing from the user's perspective, this is an important degree of freedom because the actual testing setup can use exactly those phone types which are most common in the respective environment. Device 2 will typically be a smartphone, either running industry-grade mobile network testing software, or even simple "speed test" apps. In case only specific radio access technologies shall be looked at in testing (e.g., 2G only), respective restrictions are realized by set-up properties of the smartphone or configuration of the testing software in case of industry-grade products.

In order to give a systematic approach in the context of the present Supplement, automation of testing is subdivided into modules:

– Automation of network performance testing as such, i.e., running repeated testing scenarios. Measurement data (including context information such as GNSS data) is assumed to be stored (initially) on the device.
– Automatic upload of measurement data. In order to avoid interference with testing, data upload will take place outside the time window of testing.
– Tool-supported data acquisition for DFS test cases. In the case of DFS testing (see [ITU-T P.1502]), the original approach was a generic one; readings of events on the device used for DFS testing (Device 1 of Figure 1) were written manually into lists. Those lists were then transferred into an electronic format for upload into a data-processing environment. The evolved methodology in [ITU-T P.1503] uses an app (Multi-stopwatch) to capture event

readouts from the DFS-testing device directly. Stored data is then uploaded into the data-processing infrastructure.

Fully automated network performance testing is state-of-the-art since many years and will not be discussed further. Also, automated upload of captured data will not be treated further; again, respective strategies are well-known and are assumed to be attachable building blocks to implementations of testing automation.

7 Automating the core DFS use case

The actual challenge in automation of DFS testing is related to the device running the DFS use case.

A systematic approach reveals that there are different basic types of classes of automation methods.

The first basic type of automation is a physical approach to execute actions on the phone, i.e., a literal "robot finger". Technically, this is again a well-known approach, using step-motor controlled positioning and actuators, controlled by a computer. The advantage of this approach is that it basically works with any device; no modifications to the device are required. The hardware of such an apparatus can be universal; adaptations to a particular phone model are made by software, to adapt to the geometry of the mobile device's user interface (keypad) layout.

The second basic type of controlling testing activities uses an electronic approach. This approach however needs either modifications to the target device (e.g., connecting external wires) or access to the phone's software to remotely control the phone's behaviour.

The "wiring" solution, applicable to devices which have a physical user interface such as a keypad, also works with virtually any type of device – if the extensive effort for wiring appears acceptable, which may be the case if a particular device type shall be used for a longer period of time.

The software-controlled approach requires a suitable software platform on the device. The first requirement is an operating system which basically allows this kind of access. Typically, operating systems are designed to prevent third-party control of phone behaviour, so this type of control either requires a respective physical environment (e.g., a software-development environment) or a certain degree of "hacking" the device to obtain required administrator-level privileges ("rooting").

Regardless of the type of controlling the activities on the device, executing actions is probably not enough. A typical DFS end-to-end use case is composed of several steps, and most steps can have different outcomes. For instance, after invocation of the DFS application through a USSD command, it is possible that the command does not work. E.g., due to insufficient network coverage. Even in the case of successful execution of a step, the time to completion can vary widely.

In other words, it is not enough to execute control actions. It will also be necessary to have situational awareness of the device's state, and the response to actions. Just blindly "firing away" actions, no matter if physical, mechanically or electronically, will only work in a very limited way and in a rather narrow channel of possible behaviour.

For getting information about the state of an application, there are again various possibilities. In a physical "robot finger" or externally-wired keyboard solution, essentially the only reasonable way would be a camera with respective image-processing software, unless tapping into the device's display wiring is considered as an alternative.

For software-based control, software development environments also provide access to display content or other indicators of device or application status. This leaves the task of interpreting acquired data which is, again, well within the capability of today's software technology. It should be expected, however that such a system has a significantly higher complexity than fully automated testing tools for standard use cases such as web browsing or telephony. This is because such a system will have to handle a variety of operational cases resulting from different possible responses to each step of the overall process in the test case.

In any case, even with state-of-the-art image processing, there will be a range of behaviour which is hard to capture by automated systems (unless a high effort of development is put into it). Given the fact that DFS implementations need to be rather "localized" (regionally or nationally different), and also taking into account that implementations may change from time to time, there is not an easy answer to the question where automated testing is actually breaking even with manually operated solutions.

8 Testing conditions

So far, no assumptions were made about the actual operating environment of DFS testing.

A "mobile money" implementation is assumed to always be composed of two components: the core DFS infrastructure which handles the actual transfers of money between accounts, and a mobile network which essentially provides data transport capability, no matter of the actual services used (USSD, SMS, packet data of various types).

Basically, three different types of testing are identified.

– Single-site testing: The testing system is operating always in the same place, typically with good mobile network coverage. This type of testing is useful for purposes such as basic monitoring, with a focus on the operation of the DFS subsystem.

– Stationary testing (sometimes also called "nomadic" operation). The service is used at a given location, but the overall test campaign uses a multitude of such locations, e.g., selected for a representative spectrum of mobile network coverage conditions, or for direct testing of DFS functionality in locations of interest.

– Full mobile testing: The DFS use case is executed while in motion. In this case, the spectrum of system behaviour is maximal with respect to variation of response times and the probability of breaks in transactions due to loss of connectivity.

The distinction of operational conditions is meaningful because it affects the spectrum of functional behaviour which needs to be covered by the control intelligence of automated testing. While a sequential execution of script commands without any status information from the device or the implementation may work under single-site or even stationary conditions, it is unlikely that such a system can work with reasonable robustness under highly variable mobile network conditions, even if it is assumed that data cleansing can eliminate misleading data caused by such conditions.

9 Hybrid approaches

For completeness, a hybrid class of automated testing needs to be mentioned. As outlined in [ITU-T P.1502] and [ITU-T P.1503], the proper selection of a mobile network key performance indicator (KPI) can provide at least a good estimation of the DFS service quality that can be expected, i.e., they serve as proxies for actual DFS quality.

This approach works, of course, only if it is assumed that the actual core DFS process has a certain minimum stability, i.e., degradations in service are typically not caused by flaws in DFS core processes. Also, this approach assumes that there are no significant effects resulting from time dynamics of operation via mobile networks.

Here are two examples of hybrid DFS testing solutions:

– A fully automated overall DFS testing system that is operated as a reference data source, combined with a number of mobile-network performance test systems. In this case, the cost for a fully automated system is shared by using the results for a larger number of data points.

– Fully automated mobile-network testing combined with manual but tool-assisted full DFS testing. Again, the full DFS test "channel" provides anchor data a reasonable cost-to-value ratio while bulk testing is made using mobile network KPI as proxies.

10 Special conditions for development and operation of automated DFS testing

It has already been mentioned that DFS implementations are rather highly localized. In addition, the operating area of DFS implementations may also be limited, for instance if coupled with national mobile network access. In that case, DFS testing solutions cannot be developed in an arbitrary place and be used somewhere else, which is also a factor for the cost of product development.

In addition, and independently of that, a DFS transaction has actual cost. Running a standard mobile-network performance test, e.g., web browsing, is essentially fully scalable – apart from cost for data volume, no other factor limits the number of times a particular transaction is repeated. This means that the effort to run a test system under many different conditions and with a high number of data points is small; therefore, it is likely that remaining bugs in an implementation are detected during tests.

If, in contrast, each DFS transaction costs money, either the intensity of testing has to be limited due to a given budget for development, or the cost of such tests will increase the cost of the product. This effect is enhanced when implementations change frequently, and image processing and decision processed have to be validated or updated often.

Also, repeated testing may trigger security features of DFS networks. If automated testing is used to maximize data yield, it creates a highly regular pattern of repeated behaviour, and may lead to blocking of accounts. It is of course possible to create exemptions, in co-operation with DFS operators. This may, however, not be desired from the viewpoint of objectivity as such testing entities may get a different treatment than a typical subscriber.

Last but not least, there may be regular system behaviour which limit the daily number of transactions. This happens with manual testing too, but it limits the potential gain in data productivity that automated testing can create. It is, however, not a severe problem as it can be overcome by using a larger number of accounts.

11 Overview: Types of automated testing systems and main characteristics

The following table summarizes the types of testing systems and their main properties. Please note that this table only deals with the action-creating part; see the text above for considerations on obtaining status information of mobile devices and/or applications.

Table 1 – Types of testing systems and their main properties

	Advantage	Disadvantage
Computer-controlled mechanical actuators (physical "robot finger")	Works with all types of devices (also feature phones), no modifications to the devices required	Typically large, harder to use in conditions of mobility
Hardware-wired devices with external computer control	Useful for feature phones	Not applicable/useful for smartphones; requires physical modification of devices
Software control	Relatively easy to implement	May not work with feature phones/phones not having respective operating-system functions Requires special access to device software (e.g., obtaining administrative control)

12 Summary and conclusions

A systematic look at ways and means for automation of testing DFS shows that there are different classes of systems, each with specific advantages and disadvantages. This Supplement presents an overview, and a classification scheme, for respective solutions. There is no doubt that, technically, fully automated testing is possible. Given the special conditions and circumstances, the complexity of a testing system is however assumed to be significantly higher than the complexity of automated testing of other mobile network-based services.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	**Telephone transmission quality, telephone installations, local line networks**
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Published in Switzerland
Geneva, 2023