

International Telecommunication Union

**ITU-T**

TELECOMMUNICATION  
STANDARDIZATION SECTOR  
OF ITU

**Q.3930**

(08/2012)

**SERIES Q: SWITCHING AND SIGNALLING**

Signalling requirements and protocols for the NGN –  
Testing for next generation networks

---

## **Performance testing of distributed systems – Concepts and terminology**

Recommendation ITU-T Q.3930



International  
Telecommunication  
Union

**Documents**  
Document Sharing Hub

## SWITCHING AND SIGNALLING

|                                                                                             |                      |
|---------------------------------------------------------------------------------------------|----------------------|
| SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE                                              | Q.1–Q.3              |
| INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING                                          | Q.4–Q.59             |
| FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN                                    | Q.60–Q.99            |
| CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS                                                | Q.100–Q.119          |
| SPECIFICATIONS OF SIGNALLING SYSTEMS No. 4, 5, 6, R1 AND R2                                 | Q.120–Q.499          |
| DIGITAL EXCHANGES                                                                           | Q.500–Q.599          |
| INTERWORKING OF SIGNALLING SYSTEMS                                                          | Q.600–Q.699          |
| SPECIFICATIONS OF SIGNALLING SYSTEM No. 7                                                   | Q.700–Q.799          |
| Q3 INTERFACE                                                                                | Q.800–Q.849          |
| DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1                                                  | Q.850–Q.999          |
| PUBLIC LAND MOBILE NETWORK                                                                  | Q.1000–Q.1099        |
| INTERWORKING WITH SATELLITE MOBILE SYSTEMS                                                  | Q.1100–Q.1199        |
| INTELLIGENT NETWORK                                                                         | Q.1200–Q.1699        |
| SIGNALLING REQUIREMENTS AND PROTOCOLS FOR IMT-2000                                          | Q.1700–Q.1799        |
| SPECIFICATIONS OF SIGNALLING RELATED TO BEARER INDEPENDENT CALL CONTROL (BICC)              | Q.1900–Q.1999        |
| BROADBAND ISDN                                                                              | Q.2000–Q.2999        |
| SIGNALLING REQUIREMENTS AND PROTOCOLS FOR THE NGN                                           | Q.3000–Q.3999        |
| General                                                                                     | Q.3000–Q.3029        |
| Network signalling and control functional architecture                                      | Q.3030–Q.3099        |
| Network data organization within the NGN                                                    | Q.3100–Q.3129        |
| Bearer control signalling                                                                   | Q.3130–Q.3179        |
| Signalling and control requirements and protocols to support attachment in NGN environments | Q.3200–Q.3249        |
| Resource control protocols                                                                  | Q.3300–Q.3369        |
| Service and session control protocols                                                       | Q.3400–Q.3499        |
| Service and session control protocols – supplementary services                              | Q.3600–Q.3649        |
| NGN applications                                                                            | Q.3700–Q.3849        |
| <b>Testing for next generation networks</b>                                                 | <b>Q.3900–Q.3999</b> |

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

# Recommendation ITU-T Q.3930

## Performance testing of distributed systems – Concepts and terminology

### Summary

Recommendation ITU-T Q.3930 describes terminology and concepts of performance tests with a generalized view of performance characteristics as the starting point.

What kind of characteristics are indicators of a product's performance and what kind of measurement data is captured and processed to provide relevant figures on requested performance are in this view the core of performance testing.

The Recommendation includes a set of terminology and descriptions of concepts in performance testing that could be accepted as a common base for descriptions of performance and performance tests.

### History

| Edition | Recommendation | Approval   | Study Group |
|---------|----------------|------------|-------------|
| 1.0     | ITU-T Q.3930   | 2012-08-13 | 11          |

### Keywords

Performance testing.

## 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

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

## INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation 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 Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2013

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

## Table of Contents

|                                                                   | <b>Page</b> |
|-------------------------------------------------------------------|-------------|
| 1 Scope .....                                                     | 1           |
| 1.1 Organization of the present Recommendation .....              | 1           |
| 2 References.....                                                 | 1           |
| 3 Definitions .....                                               | 1           |
| 3.1 Term defined elsewhere .....                                  | 1           |
| 3.2 Terms defined in this Recommendation.....                     | 1           |
| 4 Abbreviations and acronyms .....                                | 1           |
| 5 Conventions .....                                               | 2           |
| 6 Performance characteristics .....                               | 3           |
| 6.1 Classifying performance characteristics into categories ..... | 3           |
| 6.2 Powerfulness characteristics.....                             | 3           |
| 6.3 Reliability characteristics .....                             | 8           |
| 6.4 Efficiency characteristics.....                               | 10          |
| 7 Measured objects .....                                          | 13          |
| 7.1 Measured services .....                                       | 13          |
| 7.2 Measured components.....                                      | 13          |
| 7.3 Service concepts .....                                        | 14          |
| 7.4 Service characteristics .....                                 | 15          |
| 7.5 Service interfaces.....                                       | 19          |
| 8 Performance measurement data objectives and attributes .....    | 19          |
| 8.1 Performance metric objectives .....                           | 19          |
| 8.2 Measurement data attribute sets .....                         | 21          |
| 8.3 Processing attributes or metric types.....                    | 21          |
| 8.4 Identification attributes or metric identifiers.....          | 22          |
| 8.5 Unit attributes or metric formats .....                       | 23          |
| 8.6 Conditional attributes .....                                  | 24          |
| 9 Abstract performance metrics.....                               | 24          |
| 9.1 Abstract performance metrics and performance categories ..... | 24          |
| 9.2 Abstract powerfulness metrics .....                           | 25          |
| 9.3 Abstract reliability metrics .....                            | 28          |
| 9.4 Abstract efficiency metrics.....                              | 36          |
| 10 Performance data processing .....                              | 41          |
| 10.1 Steps in performance data processing .....                   | 41          |
| 10.2 Time series of performance data .....                        | 42          |
| 10.3 Collection and storage of raw performance data .....         | 42          |
| 10.4 Condensation and normalization of raw performance data ..... | 42          |
| 10.5 Performance data computations .....                          | 42          |

Copyright International Telecommunication Union/ITU Telecommunication Sector

|    |                                                               | <b>Page</b> |
|----|---------------------------------------------------------------|-------------|
|    | 10.6 Evaluation of performance data.....                      | 43          |
|    | 10.7 Presentation of performance data .....                   | 43          |
| 11 | General performance test concepts.....                        | 43          |
|    | 11.1 Performance tests.....                                   | 43          |
|    | 11.2 Performance tests and system life cycle phases .....     | 43          |
|    | 11.3 Performance test objectives.....                         | 45          |
|    | 11.4 Performance objectives and performance requirements..... | 46          |
|    | 11.5 Performance measurement conditions.....                  | 46          |
|    | 11.6 Performance targets .....                                | 46          |
|    | 11.7 Performance measurements standards.....                  | 46          |
|    | 11.8 Some performance test characteristics .....              | 47          |
| 12 | Performance test environment .....                            | 47          |
|    | 12.1 Test environment concepts .....                          | 47          |
|    | 12.2 System under test concepts.....                          | 48          |
|    | 12.3 Test system concepts .....                               | 50          |
| 13 | Performance test specifications .....                         | 53          |
|    | 13.1 Elements of performance test specifications .....        | 53          |
|    | 13.2 Test objectives .....                                    | 54          |
|    | 13.3 Test conditions.....                                     | 54          |
|    | 13.4 Test configurations .....                                | 56          |
|    | 13.5 Test data specifications.....                            | 57          |
|    | 13.6 Test evaluation specifications.....                      | 58          |
| 14 | Workload concepts .....                                       | 58          |
|    | 14.1 Workload set or traffic set .....                        | 58          |
|    | 14.2 Workload content .....                                   | 58          |
|    | 14.3 Workload volume .....                                    | 59          |
|    | 14.4 Load concepts .....                                      | 59          |
|    | 14.5 Workload time distribution.....                          | 59          |
|    | Bibliography.....                                             | 61          |

# Introduction

The background to the present Recommendation is that few standard specifications define any kind of performance or performance targets. A common explanation to this is that capacity issues are not a matter for standardization bodies, but for product vendors and product buyers to decide. This opinion excludes the need for definition and application of other performance characteristics such as reliability, stability, efficiency and many others.

Another more fundamental reason for writing the present Recommendation is that there are no strict definitions of what kind of characteristics should be regarded as indicators of performance.

In the absence of strict definitions in this area, one tends to talk about performance characteristics in terms of types of performance tests, such as robustness tests or availability tests. A consequence of this is that conformance testing as specified in ISO/IEC 9646-1 does not cover performance tests explicitly.

A starting point is therefore a set of terminology and descriptions of concepts in performance testing that could be accepted as a common base for discussions about performance and performance tests.



# **Recommendation ITU-T Q.3930**

## **Performance testing of distributed systems – Concepts and terminology**

### **1 Scope**

The present Recommendation describes terminology and concepts of performance tests with a generalized view of performance characteristics as the starting point.

What kind of characteristics are indicators of a product's performance and what kind of measurement data is captured and processed to provide relevant figures on requested performance is in this view the kernel of performance testing.

Methods for performance testing will consequently be guided by the requirements on expected output.

A set of following Recommendations will describe strategies, methodologies and techniques of performance testing.

#### **1.1 Organization of the present Recommendation**

The present Recommendation has two parts.

Part one is about performance characteristics and performance metrics. It contains a general view of performance characteristics followed by a general view of measured objects, or what is measured in performance tests. Part one also contains general objectives and attributes of performance data and performance metrics, i.e., a conceptual view of performance metrics that is followed by a catalogue of abstract performance metrics that covers the performance characteristics described earlier. At the end of part one is a conceptual view of performance data processing, i.e., the steps between captured performance data and presented results on performance.

Part two is about performance testing concepts. It starts with a general view of performance testing followed by a conceptual view of the test environment and finally a conceptual view of performance test specifications.

### **2 References**

None.

### **3 Definitions**

#### **3.1 Term defined elsewhere**

None.

#### **3.2 Terms defined in this Recommendation**

The core text of this Recommendation defines the terms used in performance testing.

### **4 Abbreviations and acronyms**

This Recommendation uses the following abbreviations and acronyms:

|      |                                   |
|------|-----------------------------------|
| API  | Application Programming Interface |
| CPU  | Central Processing Unit           |
| CSCF | Call Session Control Function     |

|        |                                       |
|--------|---------------------------------------|
| DHCP   | Dynamic Host Configuration Protocol   |
| DIMM   | Dual In-line Memory Module            |
| DoS    | Denial of Service                     |
| GMT    | Greenwich Mean Time                   |
| HSS    | Home Subscriber Server                |
| HTTP   | Hypertext Transfer Protocol           |
| HTTPS  | Hypertext Transfer Protocol Secure    |
| IMS    | IP Multimedia Subsystem               |
| IP     | Internet Protocol                     |
| IPsec  | Internet Protocol Security            |
| IPTV   | Internet Protocol TV                  |
| IPv4   | Internet Protocol version 4           |
| IPv6   | Internet Protocol version 6           |
| IUT    | Implementation Under Test             |
| KSR    | Kilo Service Request                  |
| MSR    | Mega Service Request                  |
| MTBF   | Mean Time Between Failures            |
| OSI    | Open Systems Interconnection          |
| SC     | Supporting Components                 |
| S-CSCF | Serving-Call Session Control Function |
| SCTP   | Stream Control Transmission Protocol  |
| SIP    | Session Initiation Protocol           |
| SMS    | Short Messaging Service               |
| SOAP   | Simple Object Access Protocol         |
| SUT    | System Under Test                     |
| TC     | Tested Components                     |
| TCP    | Transmission Control Protocol         |
| TS     | Test System                           |
| UDP    | User Datagram Protocol                |
| UE     | User Equipment                        |

## 5 Conventions

None.

## **6 Performance characteristics**

### **6.1 Classifying performance characteristics into categories**

An almost infinite number of performance characteristics can be applied on any computer system. However, measuring a complete set of performance characteristics of a system (if possible) is not only impractical, costly and time consuming, it can also be argued if it will improve or confuse the understanding of the tested systems performance.

Performance measurements are focused on obtaining figures about a selected set of performance attributes of a system.

To simplify the selection of performance characteristics for a system it is convenient to group performance characteristics about similar or related aspects of performance into performance categories and select desired performance characteristics in each category. Examples of such categories can be powerfulness, reliability or efficiency characteristics of a system. These performance categories can be exemplified by a car where:

- A powerfulness characteristic is "top speed".
- A reliability characteristic is "maintenance intervals".
- An efficiency characteristic is "mileage or fuel consumption per 100 km".

### **6.2 Powerfulness characteristics**

The powerfulness category of performance characteristics contains indicators of speed and quantity of service production. The powerfulness category can be applied on all levels from system or application level down to low-level services of different components. The powerfulness category has subcategories for responsiveness, capacity and scalability.

#### **6.2.1 Responsiveness characteristics**

##### **Time definitions**

The time to handle a service request can be split into a large number of steps, where each step causes a time delay. The responsiveness of a service request is the sum of all these time delays, the total time it takes to handle it. Every time delay falls into one of three time delay categories:

1. transmission time
2. queuing time
3. processing time

##### **Transmission time and transmission rate**

Transmission time is delays caused by transferring data related to the processing of a service request. Transmission time can be measured on any level from sending a service request to the SUT down to transmission of data between CPU and memory.

Transmission time in a network is the time to send a specific amount of data between two nodes over a network. The transmission time starts when the first bit is transmitted by the sender and ends when the last bit is received by the receiver. The transmission time is a function of the amount of data being transmitted and the transmission rate or bandwidth of the used link. A network link with a bandwidth of 100 Mbit/s has a nominal transmission rate of 12.5 Mbyte/s. However, the transport and network layer protocols used to send the data between the end nodes in a packet-switched network will decrease the nominal transmission rate by 20-40%. The resulting bandwidth is the effective transmission rate. Another factor that governs the effective transmission rate is the packet size.

## Queuing time

Queuing time are delays caused by waiting for a service of some kind related to the processing of a service request. Queuing time can, like transmission time, also be measured on multiple levels from a system service request queuing to be handled by an application server down to a process queuing for CPU time.

Every case of queuing time is caused by a mismatch between available resources and requested resources.

## Processing time

Processing time are delays caused by processing a service request. Processing time can be measured on multiple levels from sending a service request to the SUT down to application code execution time in the CPU.

## Responsiveness definitions

A responsiveness value is the sum of a large number of time elements classified as transmission time, queuing time or processing time. From a performance measurement perspective such collections of time elements do not describe the nature of measured system responsiveness. Therefore, other collections of measured time elements are used to better describe the SUT responsiveness to service requests.

Responsiveness characteristics (Figure 1) describe different kinds of service processing time delivered by a system and include:

1. response time
2. round-trip time
3. latency time
4. timeliness

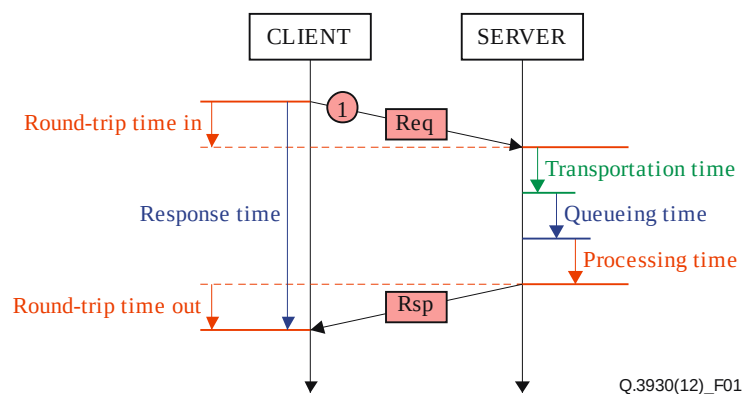


**Figure 1 – Examples of responsiveness time elements**

## Response time

Response time is the time to respond to a service request. It is usually measured from the moment the last byte of a service request is sent until the moment when the first byte of the response arrives. Response time is measured for individual types of responses to a requested service type. This applies also to performance tests with mixes of service requests.

## Round-trip time

The transfer time of data between the service requesting tool and the system under test is an important component of the response time of a service request. Round-trip time is used to separate transfer time from time spent in the SUT processing a request. Round-trip time is the time to send a

data packet to an Internet node and get a response message back, i.e., the signalling time between two nodes. Round-trip time is usually referred to as time to "ping" a node.

### Latency time

Latency time is a general term for "invisible" delays in service delivery, i.e., time spent waiting for some reaction to a service request. Latency measurements can be applied on many levels in performance tests from service request processing down to "reaction time" of individual hardware components such as a disk. In the responsiveness context any kind of latency has a negative impact on the responsiveness of a service. Here follow some examples of latency time.

*Service latency* (Figure 2) is the time to process a service request in a system. Service latency is very close to response time but does not include transmission time between requester and server.

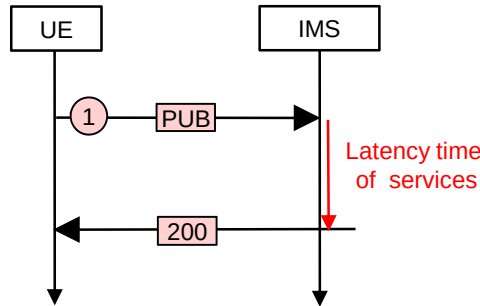


Figure 2 – Service latency

*Load level latency* (Figure 3) is the time to adjust a system to a rapid increase of incoming service requests. This can be compared to the delay from pushing the accelerator in a car to the bottom, until the car starts responding with engine working at its maximum accelerating the car.

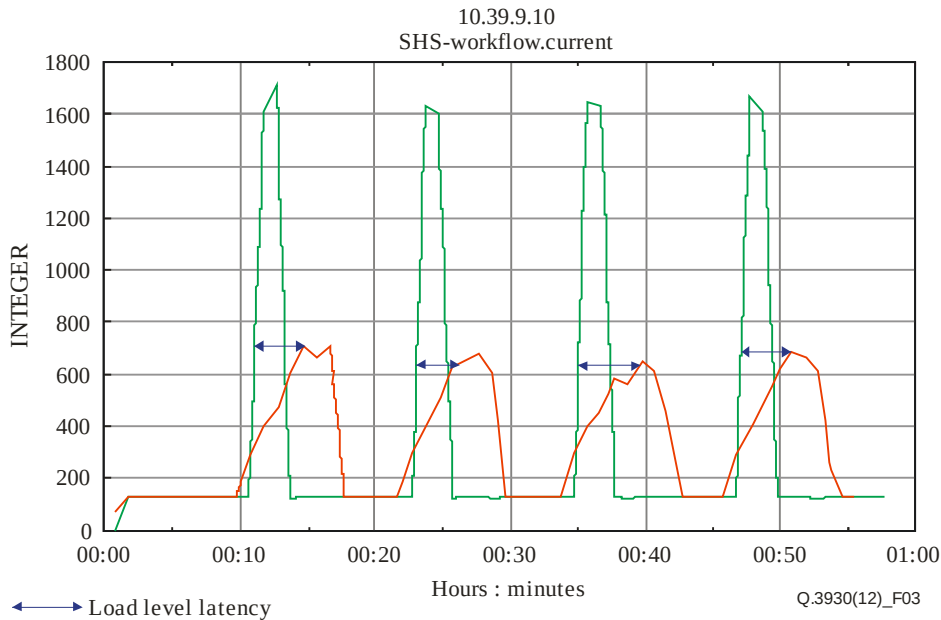


Figure 3 – Load level latency

*Latency of coupled services* (Figure 4) applies to situations where execution of a service results in execution of another "coupled" service. An example of this is the presence service in the SIP protocol. In this context, latency of coupled services is the time from update of a publication's status with a PUBLISH request, until the publication server starts sending NOTIFY messages to every active subscriber of the publication.

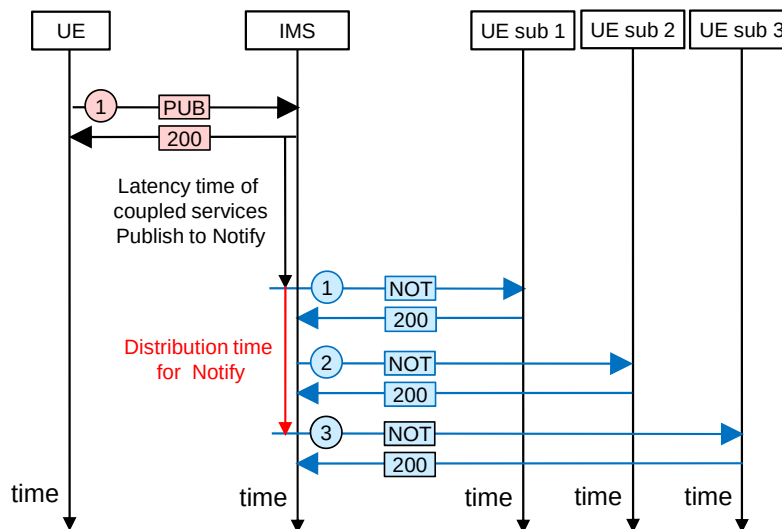


**Figure 4 – Latency of coupled service**

*Network latency* is the delay in sending data between nodes over a network.

Network latency contains several components. Each component belongs to one of two groups – static network latency or dynamic network latency.

Static network latency components are included in every data transmission and contain:

1. the transmission time to get the data across the network with the given effective transmission rate;
2. the delay time caused by the physical signalling distance between the end nodes;
3. the delay time caused by the network and transport protocol software on each node in the network chain between the end nodes; and
4. the delay time caused by security functions, such as encryption/decryption of data.

The dynamic network latency components appear occasionally and contain:

1. the queuing time for transmission due to network congestions; and
2. retransmission time to get data across the network due to poor transmission quality.

### **Timeliness or time accuracy**

Timeliness refers to measurements of delay time between received frames in a streamed service. The purpose is to measure if a data frame arrives in time to avoid noticeable disturbances in a media stream or not.

Timeliness can also be regarded as a reliability characteristic of accuracy in delivery time.

### **6.2.2 Capacity characteristics**

Capacity characteristics describe different service request volumes handled by a system including:

1. arrival capacity
2. peak capacity

3. in progress capacity
4. streaming capacity
5. throughput capacity

### **Arrival capacity**

Arrival capacity characteristics describe a system's ability to accept incoming service requests per time unit continuously on a given hardware configuration. Arrival capacity can be measured for individual services or mixes of services.

### **Peak capacity**

Peak capacity characteristics describe a system's ability to handle an overload of incoming service requests during a specified period of time on a given hardware configuration. Peak capacity can be measured for individual services or mixes of services.

### **In progress capacity**

In progress capacity characteristics describe a system's ability to handle multiple services requests concurrently on a given hardware configuration. In progress capacity can be measured for individual services or mixes of services.

Active load in progress capacity is a performance characteristic for the maximum number of concurrent services requests causing active load. Services causing active load require fast delivery and demand processing resources (CPU). Services can be stateless or stateful.

Passive load in progress capacity is a performance characteristic for the maximum number of concurrent services requests causing passive load. Services causing passive load have long duration and require no or little processing resources (CPU). Services are stateful. A typical passive load service is a pending user session. A pending user session is a prerequisite for other service requests and occupies some resources such as memory and timer functions.

### **Streaming capacity**

Streaming capacity characteristics describe a system's ability to handle multiple flows of data streams concurrently. The data streams might be multimedia streams where any variation in arrival rate of frames is critical.

### **Throughput capacity**

Throughput capacity characteristics describes a system's ability to deliver completed service requests per time unit continuously on a given hardware configuration. Throughput capacity can be measured for individual services or mixes of services.

Throughput capacity in a network describes the ability to transmit application data per time unit. The throughput capacity is limited by the bandwidth of a used transmission link. The bandwidth is a measure of the maximum number of bits that can be transmitted on a link per second. The bandwidth value is set by the hardware sending and receiving data on a link. The throughput capacity can be expressed as a percentage value of the bandwidth. Transmitted data used by the transport layer protocol and the network layer protocol in the packets surrounding the transmitted application data are not included in the throughput capacity figures. The throughput capacity will therefore always be less than 100% of the bandwidth value.

## **6.2.3 Scalability characteristics**

Scalability characteristics describe how service processing of a system can be expanded.

For distributed systems, scalability can be applied in three dimensions:

1. capacity scaling

2. distribution scaling
3. functionality scaling

### **Capacity scaling**

Capacity scaling are indicators of the relation between hardware resource increases and related service capacity increases, i.e., the SUT's ability to increase the service capacity by addition of more hardware resources to the current configuration.

### **Distribution scaling**

Distribution scaling are indicators of how service capacity and service responsiveness are affected by adding or changing the distribution nodes in a distributed system.

### **Functionality scaling**

Functionality scaling are indicators of how service capacity and service responsiveness are affected by adding or changing the functionality.

## **6.3 Reliability characteristics**

The reliability category of performance characteristics contains indicators of how predictable a system's service production is. The performance category has sub-categories for quality of service, stability, availability, robustness, recovery and correctness.

### **6.3.1 Quality-of-service characteristics**

Quality-of-service characteristics are stated requirements on service delivery performance. The requirements are usually stated as statistical values for a long period of time or a large number of service requests. Quality-of-service characteristics are therefore regarded as indicators of reliability in this context. Quality-of-service characteristics are also covered by other indicators of reliability in this clause such as:

- stability figures for acceptable/unacceptable performance
- correctness figures for correctly processed services such as transferred media codec.

Quality-of-service characteristics are also applied as measurements of network reliability. Quality-of-service characteristics of network services includes several figures such as:

- transmission failures due to network problems with following time-outs;
- transmission errors in terms of poor media quality due to corrupted data
- transmission timeliness such as poor media quality due to media packets arriving late.

### **6.3.2 Stability characteristics**

Stability characteristics are indicators of a system's ability to maintain measured performance figures for powerfulness and efficiency during service delivery regardless of time. Stability characteristics are related to measured results of powerfulness and efficiency. Stability characteristics can also be evaluations of measured performance figures versus stated figures for acceptable performance, i.e., stability as frequencies of services with acceptable/unacceptable performance. Stability figures for acceptable/unacceptable performance can also serve as performance measurements of quality of service. Stability characteristics can also be indicators of performance trends (see Figure 5), i.e., problems resulting in gradually deteriorating performance figures.

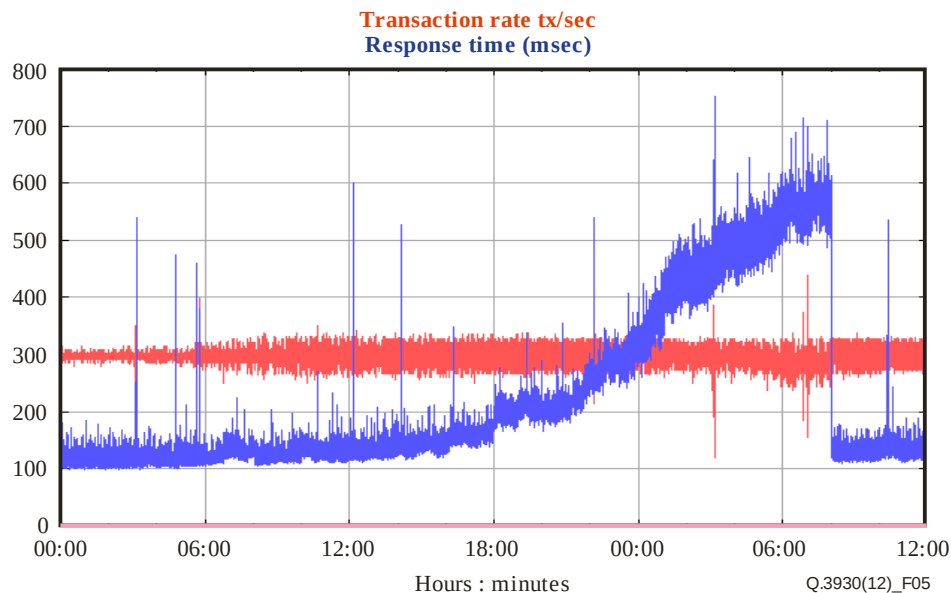


**Figure 5 – Examples of performance trends**

### 6.3.3 Availability characteristics

Availability characteristics are indicators of a system's ability to delivery services over time. Different availability characteristics are applied on hardware (physical availability) and on software (logical availability).

#### Planned and unplanned downtime

Availability characteristics cover unplanned downtime of services only. Planned downtime of services is not regarded as unavailability, however, consequences of planned downtime are usually measured in other reliability characteristics, such as recovery characteristics for time to start/restart services.

#### Physical availability characteristics

Physical availability characteristics are statistical indicators of operational time between failures for hardware equipment. Physical availability characteristics are usually expressed as uptime figures. Physical availability characteristics are usually excluded in performance measurements because they are extremely costly (broken equipment) and require testing time that is far beyond the scope of a performance test project.

Hardware equipment may include network components.

#### Logical availability characteristics

Logical availability characteristics are external measurements of frequencies of error responses to service requests due to application software problems. Error responses to bad or inadequate requests are not regarded as logical unavailability. Logical availability characteristics are usually expressed as probability figures for service delivery, but can also be expressed as uptime or frequencies for service delivery. Logical availability characteristics imply physical availability.

### 6.3.4 Robustness characteristics

Robustness characteristics are indicators of a system's services levels, i.e., services capacity and/or service responsiveness under extreme conditions. Extreme conditions can be caused internally by hardware failure or software malfunctioning, or externally by extreme peak load conditions, or by denial-of-service attacks or other malice attempts.

## **Service capacity reduction**

One consequence of extreme conditions is how the system's service capacity is affected. An ideal system is not affected at all by extreme conditions, i.e., the reduction of service level is 0%. Most likely, the service capacity is reduced to some extent, i.e., a reduction of service level in the range 1% to 99%. The worst consequence is a total stop in service production, i.e., a reduction of service level by 100%.

## **Service responsiveness deterioration**

Another consequence of extreme conditions is how the system's responsiveness deteriorates. An ideal system is not affected at all by extreme conditions, i.e., the response time is unchanged. Most likely, the service responsiveness gets worse to some extent, i.e., the response time increases. The worst possible consequence is a total stop in service production, i.e., the response time is endless, or the service request gets a time out.

### **6.3.5 Recovery characteristics**

Recovery characteristics are indicators of production disturbances from hardware or software malfunctioning. Recovery covers a large number of different operations. In this context, system recovery and service recovery are reviewed.

#### **System recovery characteristics**

System recovery characteristics are usually different measurements of time to bring a system into a fully operational state after a major production disturbance. Service recovery includes all operations from replacement of failing hardware to recovery of middleware and application components such as databases, etc.

#### **Service restart characteristics**

Service restart characteristics are usually different measurements of time to resume full service availability after system recovery procedures are complete. A more extreme version of service restart is resuming services on a back-up system or a virtual server.

### **6.3.6 Correctness characteristics**

Correctness characteristics are indicators of a system's ability to deliver correctly processed service requests under high or odd load conditions. Correctness can in this context also include streamed services, such as correctly transferred media codecs, i.e., quality-of-service characteristics for speech or multimedia transfers.

## **6.4 Efficiency characteristics**

The performance category efficiency contains different types of indicators of resource usage and resource utilization. Efficiency is measured on three levels:

1. service level
2. platform level
3. network level

On the service level, the efficiency of application service resources usage is measured. Efficiency characteristics on the service level have sub-categories for service resource usage, service resource linearity, service resource scalability and service resource bottlenecks.

On the platform level, the efficiency of platform resource distribution is measured for how well supply and demand of resources are met. Efficiency characteristics on the platform level have sub-categories for platform resource utilization, platform resource distribution and platform resource scalability.

On the network level, the efficiency of resource distribution is measured, i.e., how well does supply meet the demand for resources. Efficiency characteristics on the platform level have sub-categories for platform resource utilization, platform resource distribution and platform resource scalability.

Efficiency characteristics on the platform level also include resource utilization on network components.

#### **6.4.1 Service resource usage characteristics**

The service resource usage characteristics are indicators of the amount of resources required for processing a tested service or a mix of services. The level of service resource usage is a measure of the efficiency of the tested service or mix of services. A low level of resource usage for a service implies a more efficient implementation compared to higher level of resource usage. Measured resources can be physical (hardware resources) and logical (software resources or referenced services).

#### **6.4.2 Service resource linearity characteristics**

The service resource linearity characteristics are indicators of a system's ability to use a constant amount of resources for the production of a service regardless of the actual load level on the system. Service resource linearity is measured at different service request rates and for different services or mixes of services. Measured resources can be physical (hardware resources) and logical (software resources or referenced services).

#### **6.4.3 Service resource scalability characteristics**

The service resource scalability characteristics are indicators of a system's ability to accommodate additional platform resources for increased production of a service. Service resource scalability is for different services or mixes of services.

#### **6.4.4 Service resource bottleneck characteristics**

The service processing capacity of every system has an upper limit. If the limit is reached due to a single cause it is usually referred to as a bottleneck, since elimination of the single cause will expand the service processing capacity. The implication of a bottleneck is that available resources cannot be used efficiently to produce services. The service processing capacity will, consequently, increase significantly if an identified bottleneck is eliminated. Most systems have many bottlenecks and the most limiting bottleneck hides all the others. Therefore, the elimination of one bottleneck will only increase the service processing capacity up the limit set by the second-worse bottleneck and so on. There are five types of bottlenecks:

- configuration bottlenecks;
- processing or implementation bottlenecks;
- resource interference bottlenecks;
- design bottlenecks;
- architectural bottlenecks.

#### **Configuration bottlenecks**

Configuration bottlenecks are capacity limitations due to wrong software configurations or other similar reasons such as poor balance between hardware resources. Examples can be too few threads in a database server, or mismatch or a server that cannot use all CPU power due to lack of memory. Configuration bottleneck are usually the easiest to correct.

#### **Processing bottlenecks**

Processing or implementation bottlenecks, also called hot spots, are capacity limitations due to processing intensive spots in the code, i.e., pieces of code that are frequently executed.

## Resource interference bottlenecks

Resource interference bottlenecks, also called resource lock, are capacity limitations due to locked resources, i.e., a service puts a lock on a logical resource, such as a database lock, which in turn disables other services to execute concurrently.

## Design bottlenecks

Design bottlenecks are capacity limitations due to system design limitations (a weak design). Design bottlenecks require major efforts in redesign and implementation to get resolved, if possible.

## Architectural bottlenecks

Architecture bottlenecks, also called ultimate bottlenecks, are capacity limitations due to severely underestimated capacity requirements leading to an incorrect architecture for the system. Architecture bottlenecks can rarely be fixed. It is usually both cheaper and faster to replace a system with an architecture bottleneck. A typical example of architecture bottlenecks is an old single-threaded application that when moved to a modern multi-CPU multi-core hardware system, cannot utilize the additional processing resources.

### 6.4.5 Platform resource utilization characteristics

The platform resource utilization characteristics are indicators of to what level available platform resources can be utilized for production of a service or a mix of services. Platform resource utilization can be applied on an individual resource or a mix of resources.

The highest possible value for platform resource utilization is of course 100% of all resources used. In reality, as one hardware resource is used to its maximum there are still other types of resources unused. The difference between the least and the most utilized resource may be an indicator of a configuration bottleneck. The bigger the difference the worse is the bottleneck.

Platform resource utilization is measured on a system with a fixed amount of resources. The platform resource scalability characteristics are indicators of platform resource utilization for additional resources.

Efficiency characteristics for network resource utilization are slightly different measures with a different purpose. Efficient utilization of network bandwidth can be measured as a percentage of the effective transmission rate on a network component (the percentage of the nominal bandwidth used for successful transmission of application data).

An increase of the packet size will reduce the ratio between transmitted data and overhead data (data in a packet used by transport and network protocols to get the application data across the network) and consequently increase the effective transmission rate, i.e., lower the overhead data as a percentage of transmitted application data. However, an increase of the packet size also increases the probability of a transmission error followed by a retransmission of a failed packet. Therefore, the effective transmission rate must be reduced by the transmission rate spent on retransmitting data. From a certain point, any increase of packet size will have a negative impact on the effective transmission rate.

Another factor is the impact of the packet size on the packet switching capacity, which in turn is related to network congestion problems. With smaller packet sizes the amount of switched packets increase per time unit, which improves the likelihood of receiving data in time. Smaller packets can also be more evenly distributed across alternative network links and thereby shorten and reduce the consequences of network congestions.

### 6.4.6 Platform resource distribution characteristics

The platform resource distribution characteristics are indicators of how fast and evenly platform resources are distributed to requesting services. The platform resource distribution can be measured for an individual service or a mix of services.

A system with poor resource distribution runs out of one type of resource while there are still plenty of other resources. A system with poor resource distribution will also be less capable of handling sudden resource shortage situations following system resource outages. The platform resource distribution characteristics discussed here are of two kinds:

1. demand driven resource distribution
2. outage driven resource distribution

Demand-driven resource distribution characteristics are indicators of a system's ability to distribute available resources to demanding service requests.

Outage-driven resource distribution characteristics are indicators of a system's ability to redistribute available resources to demanding service requests after various outage situations. The objectives of outage driven resource distribution is to minimize the effects of various outage situations.

#### **6.4.7 Platform resource scalability characteristics**

The platform resource scalability characteristics are indicators of to what level additional system resources can be utilized for production of a service or a mix of services, i.e., resource utilization applied on additional resources. Platform resource scalability can be measured for an individual service or a mix of services.

The highest possible value for platform resource scalability is of course 100% of additional resources. However, there are some limitations to what is possible to reach set by the platform resource utilization, measured before the addition of resources. Few systems have a perfectly linear platform resource scalability. The effect of adding more resources of some kind is in reality limited by the capacity of other related resources. For example, the effect of adding more processing power and memory to a system is in reality limited by the transmission capacity between the CPU and memory.

## **7 Measured objects**

### **7.1 Measured services**

When requesting information about the performance of a car figures for top speed, acceleration, maximum load, mileage, service intervals, etc., are usually provided.

The performance figures apply to the tested car as a whole, i.e., on system level or top level of the tested system. However, the measured performance of a car is a result of the car design and the performance of the various components of the car involved in producing its services. Examples of the components of a car contributing to measured powerfulness values are the performance of the engine, the transmission system, the electrical system, etc. To design the performance of the car we need to measure and evaluate the performance of its components and how the components interact.

A similar approach can be applied on a computer system. At the system (application) level the performance of system service delivery is measured, such as transaction processing capacity, or response time of various services, etc.

### **7.2 Measured components**

Similar to a car, the measured performance of a computer system service is not atomic, but the end result of the performance of many levels of processing services or components. The performance of application services depends on the performance of requested middleware services. The performance of middleware services depends on requested operating system services. The performance of operating system services depends on the performance of requested hardware components services, etc.

The performance of each component involved in delivering an application service is not required to measure the performance of an application service. However, performance is built from inside out, i.e., to design an application that meets defined performance requirements, the measure and control of the performance of all components is required.

### **7.3 Service concepts**

A distributed system provides its services to users over a published interface. If a service is general enough it can be used as a shared service by multiple applications. Access to a service in a distributed system is open to any client that has the authority to use the service and is authenticated as a valid user. The rationale of this concept is reuse of software as an online service.

#### **7.3.1 Service and component performance**

An application service is normally resolved by a set of internal services.

The measured performance of a system service is the aggregated result of all components (hardware and software) involved in, and contributing to, the results.

The performance of an application service depends on the performance of requested middleware services. The performance of the middleware services depend on requested operating system services. The performance of operating system services depend on the performance of requested hardware components services, etc. The track can basically be followed down to execution of CPU instructions.

#### **7.3.2 Service topology and topology performance**

The service topology describes how an application service is dependent on other application services to resolve its task.

Performance tests of service topology focus on the responsiveness of distributed services processing components. Performance tests of service topology cover, for example:

- tests of latency in accessing distributed services;
- tests of capacity in accessing distributed services.

A distributed system is not only built on several layers of services, but each layer of services may also be distributed across a large number of system components (servers).

The system topology describes how the system components are interconnected and the requirements to traverse the system between any two components.

For example (see Figure 6) the registration of an IMS user is initiated by the UE sending a REGISTER request on the IMS Gm interface (SIP) to the user's servicing CSCF (S-CSCF). The S-CSCF in turn requests services from the operators HSS to identify and authenticate the user and set up secured IPsec channels to the UE. This is achieved by sending a request on the IMS Cx interface (Diameter) to the HSS.