



Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE:
INTERNATIONAL:
FAX:
E-MAIL:

Cambridge (01223) 515010
+44 1223 515010
+44 1223 359779
apm@ansa.co.uk

Training

ANSAwise - Service Quality in Distributed Systems

Chris Mayers

Abstract

Organizations need to be able to supply services world-wide across diverse (heterogeneous) networks. The services may require end-to-end guarantees with various levels of service quality. Users wish to purchase services with appropriate levels of service quality, and use them in their applications.

End-to-end guarantees entail not just the ability to support them within the applications services, the engineering infrastructure, and the communications protocols, but also to coordinate, manage, and federate these guarantees.

This module of the ANSAwise training programme explains how Quality-of-Service (QoS) issues are relevant to distributed systems, and gives some examples of typical applications.

This module does not cover QoS Engineering (including bindings); this is to be covered in a later module.

APM.1321.01

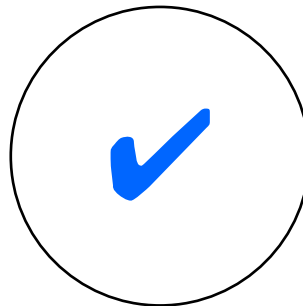
Approved
Briefing Note

21st October 1994

Distribution:
Supersedes:
Superseded by:



Service Quality in Distributed Systems





In this session

- *Show how quality-of-service (QoS) allows service quality guarantees*
- *Explain why applications will need these guarantees*
- *Show how ODP provides a framework for managing these guarantees*



Service Quality - The Customer's Need

- *In open markets, products and services are differentiated by quality*
 - different quality for a different price
- *A single service may offer a choice of quality-of-service*
 - the choices may vary over time
 - the price may vary too
- *In open markets, the choices of quality/price vary rapidly*
 - This means that dynamic QoS negotiation is necessary



Dynamic Quality-of-Service Negotiation

- *This benefits both service providers and service users*
 - service providers can optimize their use of resources
 - service users can choose how much they want to pay
 - service providers can offer QoS guarantees

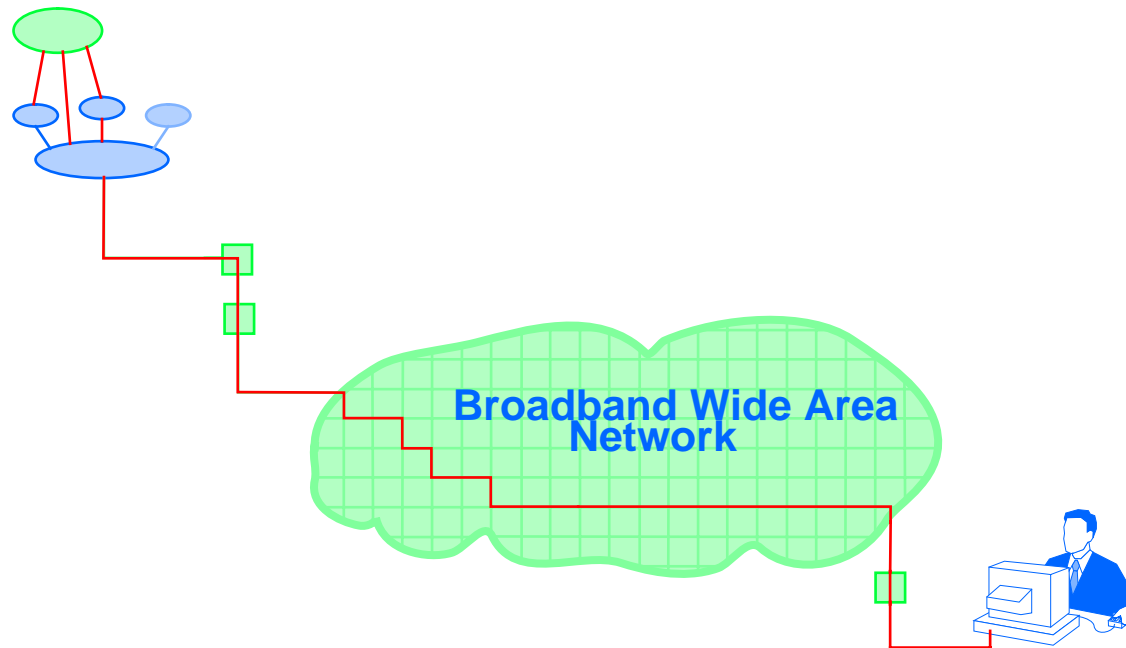


Best-Effort versus Guaranteed Service

- *Two different kinds of offer*
 - best-effort (whatever is available at the time)
 - guaranteed (minimum QoS)
- *Why not always offer guarantees? Because they have overheads*
 - guarantees require pre-allocation of resources
 - guarantees require monitoring and measurement
 - the service provider has to insure itself against failure
- *Best-effort service will continue to have a place*

How guarantees affect distributed systems

- *The guarantees must be end-to-end guarantees...*



- *... possibly over a wide-area network*



End-to-end guarantees

- *Over a network (even just a LAN), resources are shared...*
 - ...bandwidth, buffers, processors,...
- *Maintaining guarantees entails pre-allocating resources*
 - at least for the period of the guarantee
- *The end-to-end route may pass through different service providers*
 - it must be possible to *federate* the guarantees
- *A dedicated infrastructure is probably impractical*



Quality-of-Service - The Application's Need

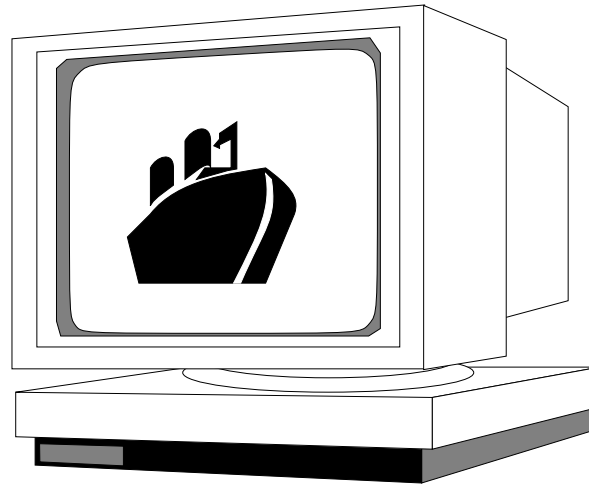
- *Some applications rely on guarantees, for example*
 - for continuous media (video, audio)
 - for transaction processing (open OLTP)



Different Applications, Different Guarantees

- ***Continuous media***
 - low latency and high bandwidth
 - bounded jitter
- ***Transaction processing***
 - bounded response time
- ***Future applications may require both these elements, and more***

Travel reservation system



- *Customers watch a video of the holiday destination...*
- *... and make the booking as well*



Service Quality in the real world

- *You are making a long train journey. What do you expect the railway company to provide you? Split your requirements into three groups*
 - functionality
 - guaranteed QoS
 - best effort QoS
- *Try thinking of a journey when your expectations weren't met*



Your notes

Get ready to discuss this



What is covered by Quality-of-Service?

- *Broadly speaking, QoS covers all the 'non-functional' characteristics of an interface*
 - *functionality is the **computational** part*
 - *QoS is everything else*
- *QoS characteristics are English words ending in “-ility” or “-ness”*
 - *availability*
 - *timeliness*



What are all these are characteristics?

- *Different people use the same word to mean different things*
 - the differences are important
- *Please write down what you think “availability” means*



What does “dependability” mean?

- *rate of occurrence of failures?*
- *probability of failure on demand?*
- *probability of failure-free survival of mission?*
- *availability?*

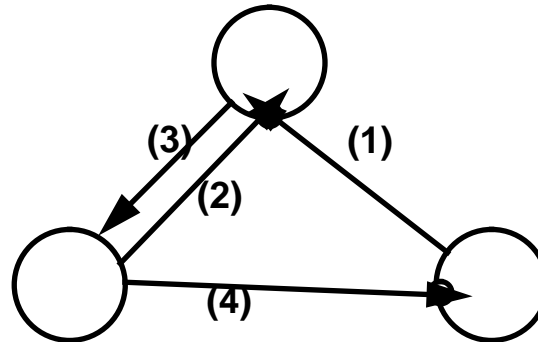
... and there are many other characteristics



Federating guarantees

- *At a boundary between service providers, it must be possible to match up QoS guarantees*
- *This requires agreements on*
 - *a standard framework for QoS guarantees*
 - *standard meanings for QoS characteristics*
- *Standard meanings are an ODP information viewpoint issue*
 - *these meanings must match customer needs for Service Quality*

Quality-of-service matching and negotiation



- *The ANSA/ODP trading model is the basis for QoS matching*
 - but needs extending to handle end-to-end negotiation
- *QoS negotiation will be a separate service*
- *Negotiation can take place at any time*
 - even during a connection
- *The trading model already allows for federation*

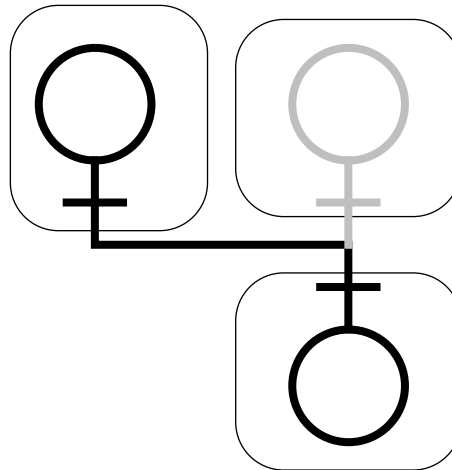


Simplifying Service Quality

- *QoS must be easy for the applications designer to use...*
 - ...even though the QoS infrastructure may be complex.
- *Two familiar approaches help*
 - declarative QoS specifications (used in trading)
 - transparencies

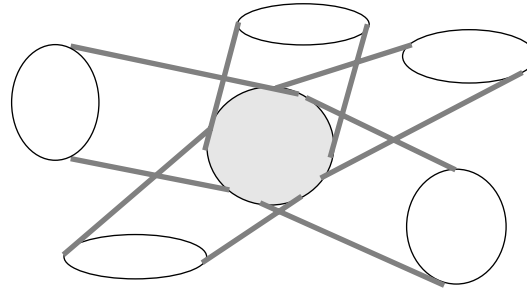
Transparencies and Service Quality

- *When services migrate, their QoS may change....*



- *... but this can all be hidden from the application by a transparency*

Quality of Service and the ODP Viewpoints



- *Enterprise: correspondence to customer needs*
- *Information: meaning of QoS terms*
- *Computational: associating QoS with computational entities*
- *Engineering: resource allocation and management*
- *Technology: selection of hardware and operating systems that can provide the necessary guarantees*



A sensitive subject - the fifth force

- *Computers*
- *Communications*
- *Consumers*
- *Content*
-



Measurement

- *Billing must account for quality-of-service*
- *Quality-of-service must be measurable*
 - measuring the quality that was actually delivered
- *We must measure characteristics that we cannot control directly*
 - for example, MTBF (mean time between failures)



Why is Quality-of-Service important now?

- *Applications and infrastructure handle QoS in ad-hoc ways*
 - different layers do things differently
- *New application demands (multimedia, open OLTP)*
- *New technologies with wide-spectrum capabilities*
 - Asynchronous Transfer Mode (ATM)
 - Mobile networking

...and a perceived customer demand



An Approach to Quality-of-Service

- ***As service provider***
 - is there a market for a service with guaranteed QoS?
 - if so, which levels of QoS should be provided?
 - how will monitoring and measurement be performed?
 - what about the services that you use?

- ***As service user***
 - does your application require guaranteed QoS, or is best effort sufficient?
 - do alternative providers offer a choice of QoS?
 - will you monitor the QoS yourself?
 - what about the services that you provide?

- ***As service provider or user***
 - which new technologies do you wish to exploit?



Summary

- *End-to-end processes need end-to-end guarantees*
- *The infrastructure must support these guarantees*
- *Quality-of-services guarantees should be declarative*
 - *hiding the complexity from the applications*
- *For more on service quality and quality-of-service*
 - *on continuous media, see [Integrating Multimedia into the ANSA Architecture \(TR.028\)](#)*



An example definition - Dependability

- *“The property of a system that allows reliance to be justifiably placed on the services it delivers”*
- *Defined in terms of five non-functional properties*
 - *availability: readiness of usage*
 - *reliability: continuity of service*
 - *integrity: maintaining the consistency of data*
 - *safety: avoidance of catastrophic consequences*
 - *security: prevention of unauthorized disclosure (or handling) of information*