PREDICTABLE
NETWORK
SOLUTIONS

Performance Contracts in SDN Systems

May 2017

Published in IEEE Softwarization - May 2017

Abstract

SDN virtualizes connectivity and access to the underlying bearers. This enables more variety of routes and new ways to share the bearers to meet customer demands at lower cost. However customers will need assurances about the fitness for purpose of the delivered service for their critical applications. This requires new ways to quantify their requirements and to measure the delivered service that go beyond simple notions of bandwidth/capacity.

Dependability of network performance

How can users of a network have confidence that it is delivering satisfactory performance? In a traditional context, where network configuration changes only occasionally, this can be achieved through testing. With SDN, however, where the network configuration is highly dynamic, this approach has limited applicability. Either tests must be run very frequently, thereby imposing load on the network, or the user has to accept that most network configurations are untested. This is an issue for both the SDN user and the SDN supplier: how does the user know what capacity and SLA to ask for; and how can the supplier decide what level of statistical multiplexing they should target?

Therefore, to maintain user trust, the use of SDN requires a much more proactive approach to quantification and qualification of performance, which needs to be embodied in service contracts. This needs to be based on a more precise definition of performance, including how this impacts the user.

The nature of ‘performance’

‘Performance’ is typically considered as a positive attribute of a system. However, a ‘perfect’ system would be one that always responds without error, failure or delay; real systems always fall short of this ideal, so we can say that the *quality* of their response is *impaired* relative to the ideal (such ‘quality impairment’ is thus a privation).

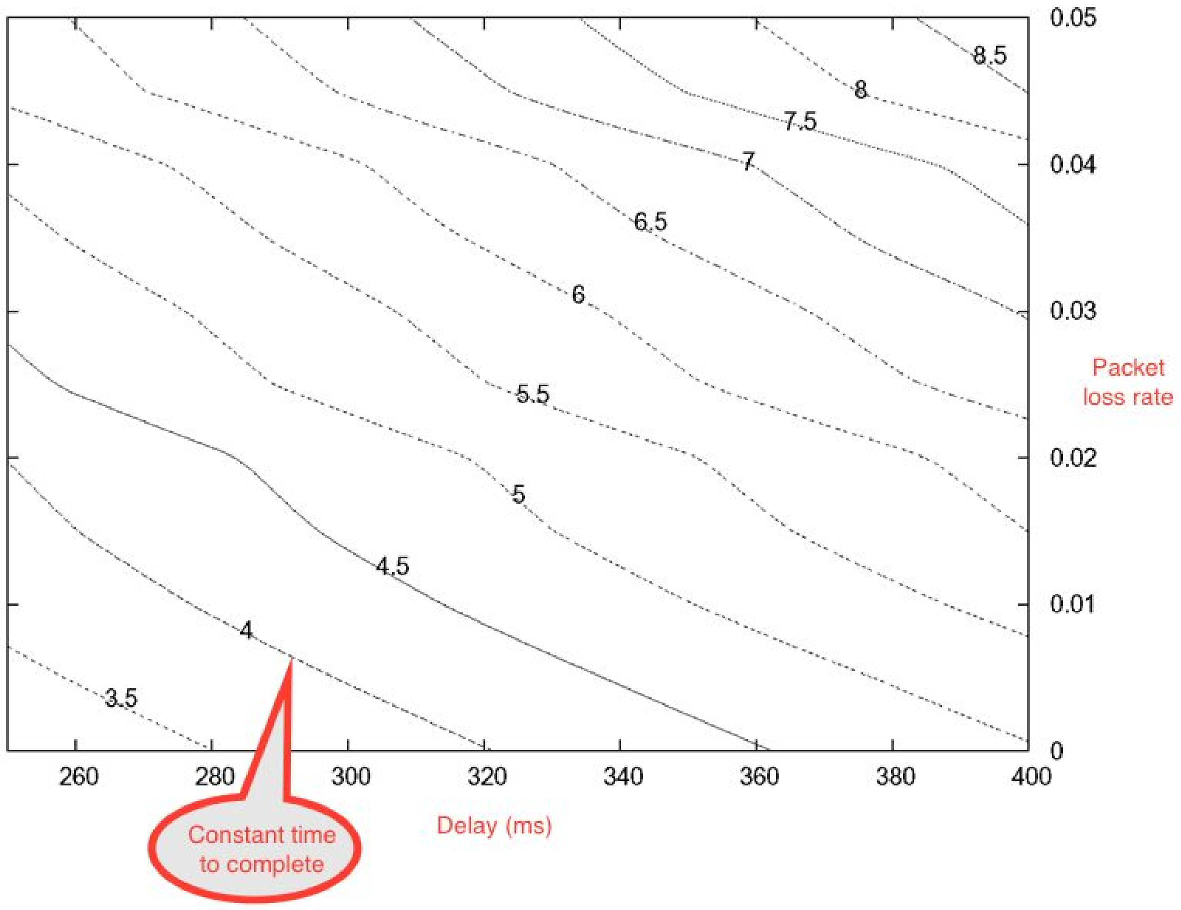
Performance has many constraints, starting from the fact that doing anything takes time and uses resources. Geographical distance defines the minimum possible delay; communication technology sets limits on the time to send a packet and the total transmission capacity; and sharing of resources limits the capacity available to any one stream of packets. The design, technology and deployment of a communications network (which is made more dynamic by SDN) sets the parameters for a best-case (minimum delay, minimal loss) performance at a given capacity. This is what the network ‘supplies’, and this supply is then shared between all the users and uses of the network. This sharing can only reduce the performance and/or the capacity for any individual application/service. Networks share resources statistically and so the resulting quality impairment is also statistical.

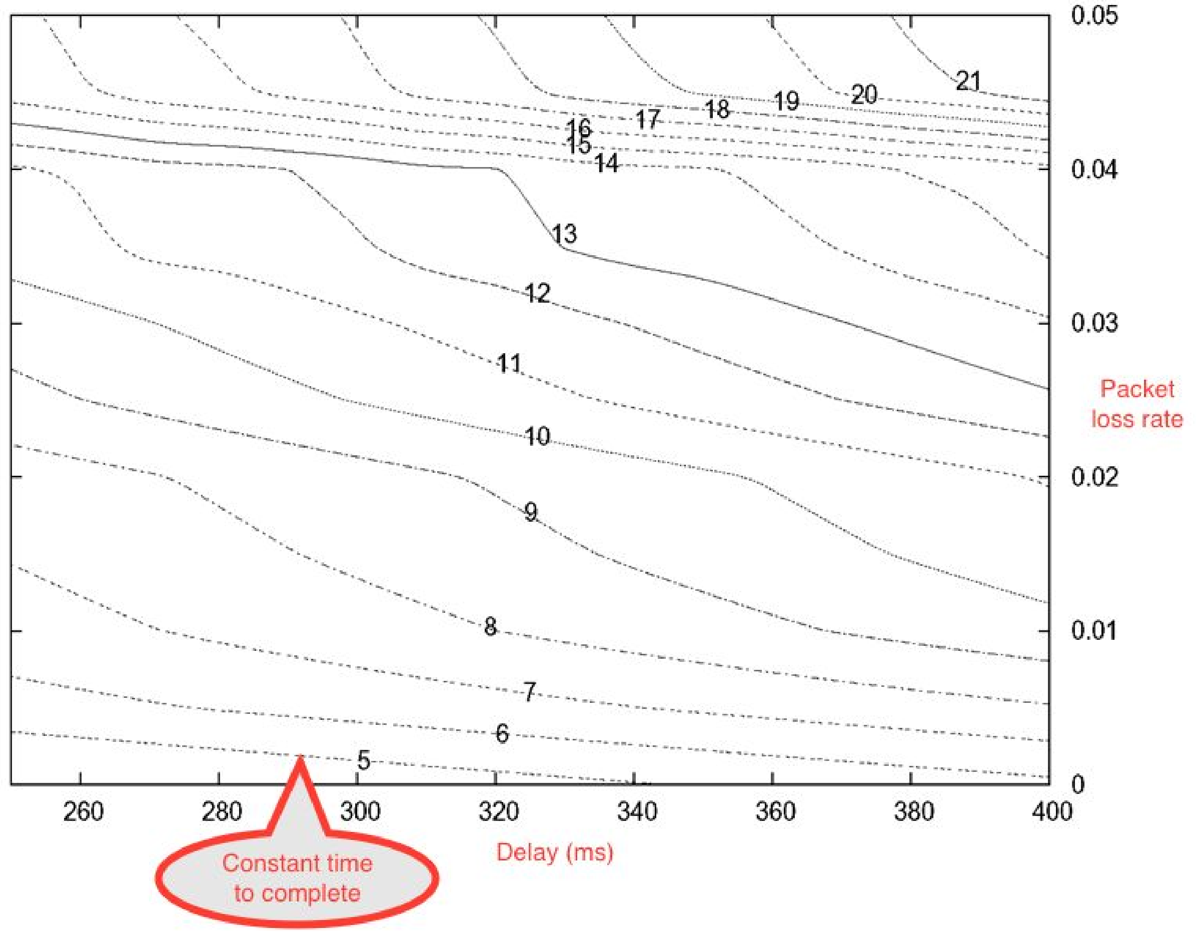
Since quality impairment is statistical, 100% delivery is unachievable - and the underlying bearers only have ‘N nines’ availability in any case. Performance thus needs to be considered statistically - but not as averages, as we shall see below.

Performance Requirements

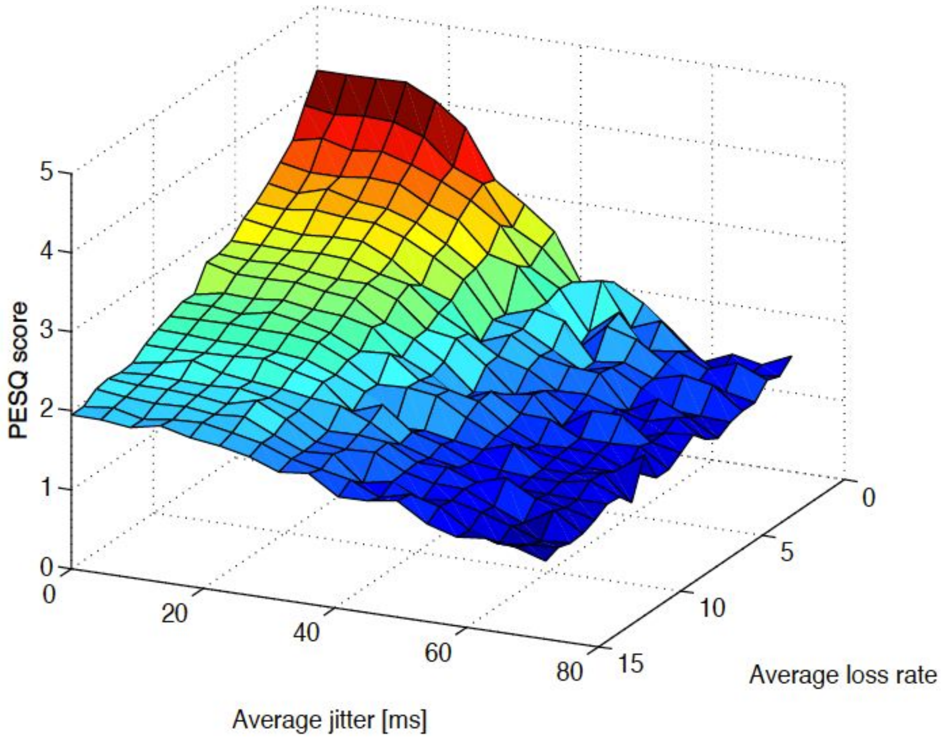
Typical audio impairments that can affect a telephone call (such as noise, distortion and echo) are familiar; for the telephone call to be fit for purpose, all of these must be sufficiently small. Analogously, we introduce a new term, called ‘quality attenuation’ and written ‘ΔQ’, which is a statistical measure of the impairment of the translocation of a stream of packets when crossing a network. This impairment must be sufficiently bounded for an application to deliver fit-for-purpose outcomes; moreover, the layering of network protocols isolates the application from any other aspect of the packet transport. This is such an important point that it is worth repeating: the great achievement of network and protocol design (up to and including SDN) has been to hide completely all the complexities of transmission over different media, routing decisions, fragmentation and so forth, and leave the application with only one thing to worry about with respect to the network: the impairment that its packet streams experience, ∆Q.

The impact of any given level of impairment depends on the particular protocols that an application uses and on the time-sensitivity of its delivered outcomes. For example, here is a graphical representation of the effect of varying levels of delay and loss (parameterised by the mean of uniform distributions) on the means and 95th centile of the time to complete a 30kB http 1.0 transfer (contours are labelled in seconds):





Here, on the other hand, is a plot (measured by a researcher at CERN) of the PESQ score of an H.264 audio decoder as a function of the loss rate and jitter of the packet stream:

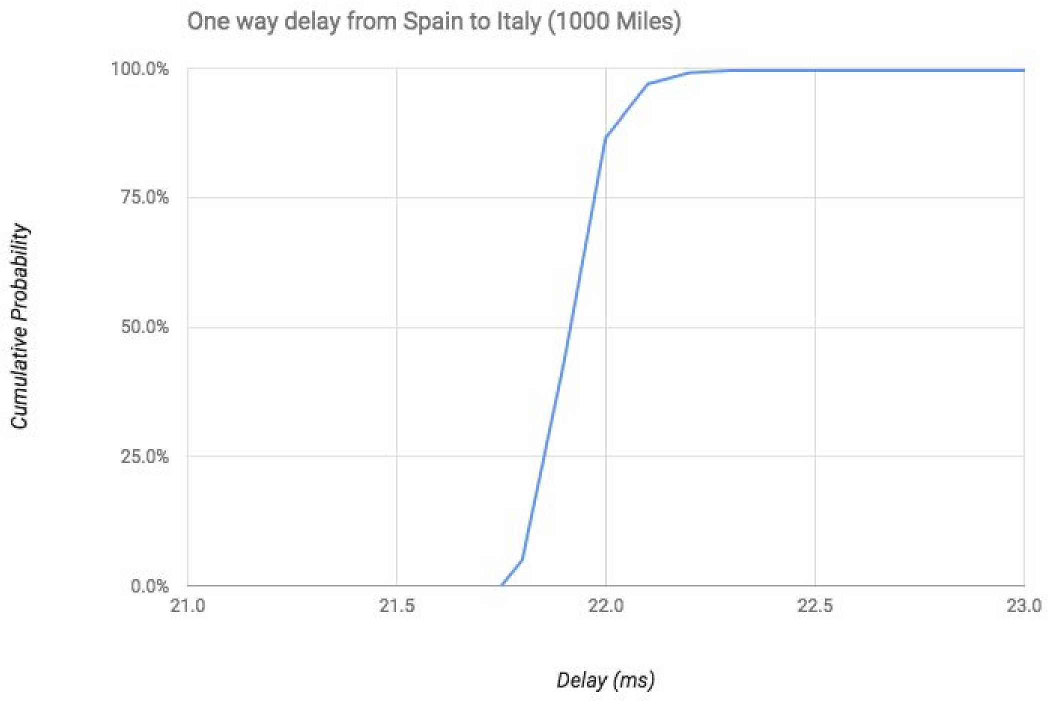


VoIP is sensitive to jitter whereas HTTP is not; how can we characterise the delivered service in a way that applies to both? In principle, the answer is to use the complete distribution of packet delays (from which the mean delay, jitter or any other measure can be derived), together with information about packet loss, corruption, out of sequence delivery and so forth. While this works mathematically, applying this practically requires some further ideas, which we discuss in the next section.

Quantifying performance

Measuring ∆Q

One approach to practically measure ∆Q is to inject a low-rate stream of test packets and observe their transit times at various observation points. By keeping the rate of the stream low it is unlikely to materially affect the instantaneous utilisation of any resources. The distribution of packet delays (and the proportion of losses) provides a measure of ∆Q. On the assumption that packets do not overtake each other and that the test packets are not given any special treatment, measuring the ∆Q of the test stream provides a sample of the ∆Q of all packets following the same path. The figure below shows the cumulative distribution function of just such measurements made between two points 1000 miles apart, connected by core network links:



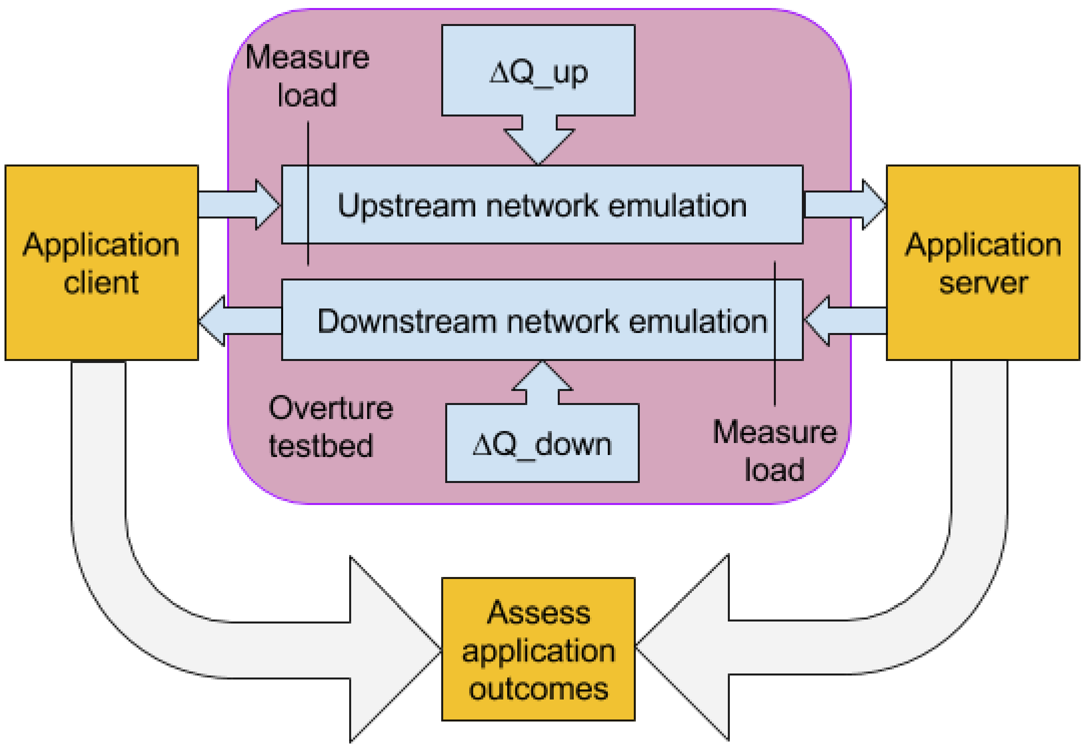
Such data has structure that can be unpacked to expose a great deal about the operation of the network path, but the point here is that the abstract idea of ∆Q can be realised in practice, and provides a single measure of network performance that can be related to the likely behaviour of any application sending packets along the same path.

Quantifying the impact of ΔQ

So, ΔQ can be measured, but how can we establish what ΔQ is or is not acceptable for any particular application? Fortunately, the UK public body InnovateUK has funded the development of a generic test environment in a project called Overture (project no. TP710826). This is illustrated in the following figure, which shows client and server components of an application (more complex configurations can also be used) exchanging packets through the testbed. The upstream and downstream ΔQ can be controlled (and varied) independently; combining this with a measure of how acceptable the application performance is (that may be subjective) sets bounds on the ΔQ such an application can tolerate. To completely characterise the application performance impact of network ΔQ as shown above for http and VoIP is a daunting task, but establishing a broad envelope of acceptability is quite straightforward. This can be captured in a fairly simple specification, such as:

- 50% delivered within 5ms
- 95% delivered within 10ms
- 99% delivered within 15ms
- 1% loss acceptable

Overture testbed setup

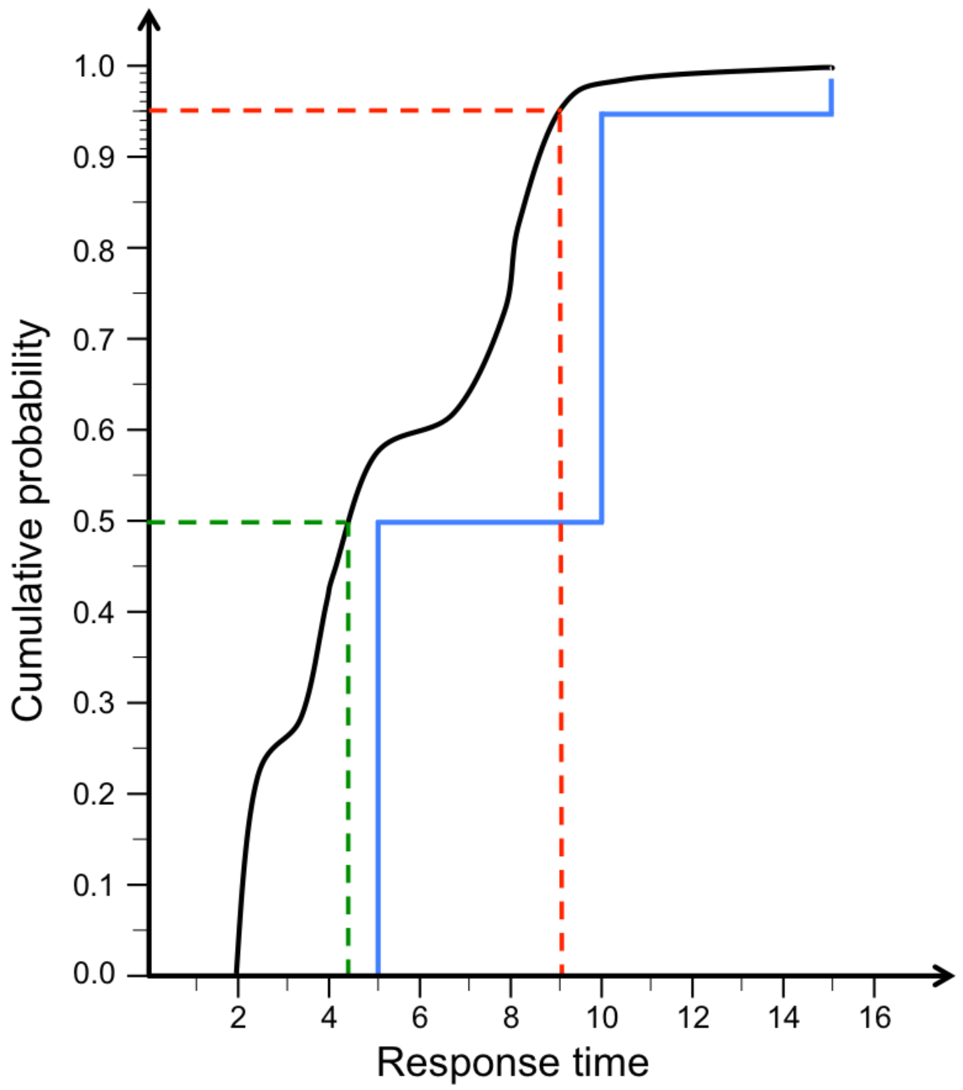


At the same time, the load imposed on the network by the application can also be measured, which is important for completing the performance contract, as described below. Note that 'load' is also characterised by a distribution, not just an average, which is important for the ability of the network to carry this load and still deliver acceptable ΔQ, for example deciding how much capacity to allocate in excess of the mean.

Describing the contract

QTA

Given the information above, we can formulate a technical specification of the relationship between application load and network performance. We call this a ‘Quantitative Timeliness Agreement’ or ‘QTA’ which consists of a set of constraints on the distribution of load and the distribution of loss and delay. Considering the distribution of loss and delay, combining an application requirement as described above with a measure of delivered performance at the SDN service layer would produce a pair of cdfs such as shown below:



In this case we can see that the measured performance is unambiguously ‘better’ than the requirement (for example, 50% of packets are delivered within about 4.2ms whereas the requirement was within 5); this is the criterion for the network to have ‘met’ the QTA. The constraints on the application load on the network can be managed similarly, and so we have clear and unambiguous criteria for conformance on both sides. Note that these can be extended to measures of by how much a requirement has been met (or not met), but to a first approximation a binary characterisation is sufficient.

Relationship between QTA and SLA

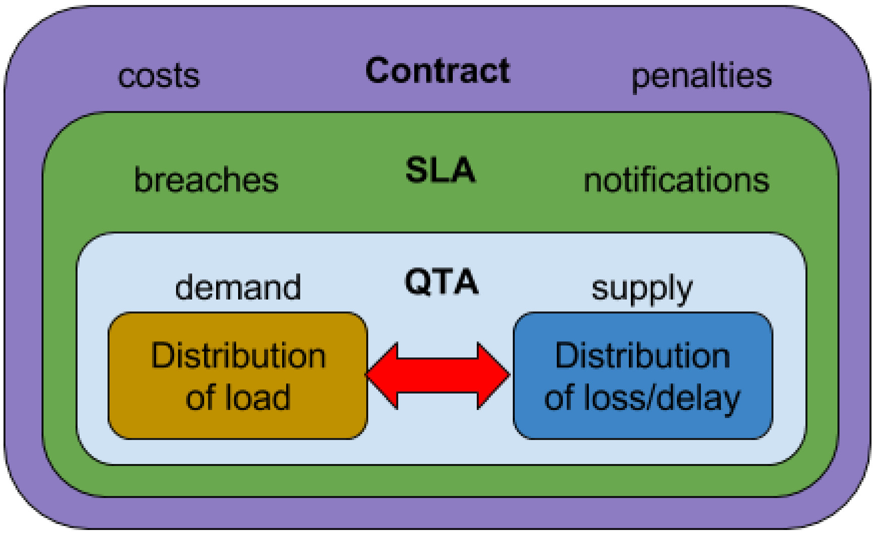
The QTA captures the measurable aspects of the application demand and the network supply; the SLA is the framework surrounding it.

		Offered Load	
		In Specification	Out of Specification
Delivered Supply	In Specification	✓	The network is exceeding its contracted requirements - maybe an alarm condition
	Out of Specification	Failing to meet the ∆Q bounds on delivery places QTA in breach.	If load is excessive there is no requirement it should be delivered within ∆Q bounds

The SLA will specify under what circumstances the QTA may be breached, some of which might involve prior notification, and may place limits on how often and for how long this can happen. This provides the application with a clear measure of the technical risk it faces from the network, which it needs to either mitigate or propagate upwards.

Relationship between SLA and contract

The contract captures all remaining issues, including the consequences of breaching the SLA. It is important to note that there is an asymmetry in risk between the supplier and the consumer of the service: the supplier might suffer a financial penalty for non-delivery, but the consequences for the application might be beyond such remedies, if it relates to safety of life for example.



This gives an overall picture of a performance contract as in the figure above. SDN systems need to evolve to deliver such contracts dynamically and efficiently to provide enhanced value to applications and services of the future.