Measuring Network Impact on Application Outcomes Using Quality Attenuation

Neil Davies, Peter Thompson
Predictable Network Solutions Ltd
`neil.davies@pnsol.com, peter.thompson@pnsol.com`

Gavin Young, Jonathan Newton
Vodafone Group PLC
`gavin.young2@vodafone.com`
`jonathan.newton@vodafone.com`

Bjørn Ivar Teigen, Magnus Olden
Domos AS
`bjorn@domos.no, magnus@domos.no`

September 2021

1 Summary

Starting from an objective view of how the network impacts a distributed application's performance, we develop a quantitative measure of network quality based on the notion of 'observations' that is tractable and composable. This guides practical measurement implementations based on constructing observable events.

The mathematical properties of this method produce high spatial and temporal localisation from low-impact measurements that can be run continuously. This can be used to both assure end-to-end conformance and localize 'breaches' to specific network segments, enabling both service and contractual assurance. In laboratory and field trial settings, it can be used for testing configurations, properties of firmware, differences between different access technologies, etc.

After using this approach in several successful technical engagements, the authors presented the general concepts to the Broadband Forum in late 2018. The work was developed as the QED (Quality of Experienced Delivered) project within the PEAT (Performance, Experience and Application Testing) work stream, culminating in an extensive study document. A series of technical documents [12, 10, 5, 13] followed from this, in particular TR-452.1 [10] from September 2020. This standard has enabled multiple implementations, deployed to solve real problems. Several vendor implementations were demonstrated at Broadband World Forum in 2019. The initiative is ongoing.

2 Introduction

The intrinsic value of the Internet is in delivering outcomes: network-mediated activities/interactions that combine communication and computation to provide, among other things, 'information at a distance'. Lasting societal value arises from delivering such outcomes in a way that is 'fit-for-purpose', consistent and cost-effective. Being fit-for-purpose does not mean absolute perfection, but it does demand making bad experiences *sufficiently rare.*

Without an appropriate measurement framework these goals are merely aspirations. Robust quantification is a prerequisite for a framework that can deliver satisfactory outcomes. This document describes a framework for defining end-to-end network quality in a way that is relatable to high-level outcomes. We point to relevant academic work, ongoing standardization efforts, and current commercial deployments.

The basis of this framework is the concept of ‘Quality Attenuation’, which has been standardized by the Broadband Forum [10]. The requisite measurements can leverage existing IETF protocols such as STAMP and TWAMP [4, 2]. Standardization offers the prospect of multi-vendor interoperability.

This framework has emerged from many years of practical engagements with large-scale/complex distributed systems [10, 3, 8, 7]. During its development, consideration has been given to mathematical soundness, metrology and practical deployment.

3 Embracing imperfection

From the perspective of an end-user, the network is simply an enabler of particular distributed outcomes. For the network, delivering ‘good performance’ is about making bad user experiences rare; those in which the particular application of interest fails to perform appropriately, adversely affecting the user experience (UX). Insufficient capacity will generate bad experiences, but sufficient capacity does not guarantee that they will be absent.

A perfect network would have zero packet delay and zero packet loss, i.e. zero defects. In real networks, all packets are delayed and some may get lost, even at low loads, hence quality is ‘attenuated’ compared to the perfect network outcome. This is the basis for Quality Attenuation measurement and analysis: to statistically characterize the real-world degradation of quality and hence quantify the ‘delta’ from a perfect network, which we notate as ΔQ. It can be thought of as either the probability distribution of what might happen to a packet transmitted at a particular moment from source A to destination B or as the statistical properties of a stream of such packets (each of which corresponds to a sample from the instantaneous distribution).

In capturing the deviation from ideal behavior, ΔQ incorporates both delay (a continuous random variable) and exceptions/failures/losses (discrete variables). This can be represented mathematically using improper random variables (IRVs), whose total probability is less than one. If we write ΔQ(x) for the probability that an outcome occurs in a time $t \leq x$, then we can define the intangible mass of such an IRV as $1 - \lim_{x\to\infty} \Delta Q(x)$, which encodes the probability of exception/failure/loss. This is illustrated by figure 1, showing the cumulative distribution function (CDF) of an IRV (with arbitrary time units).

We can define a partial order on such variables, in which the ‘smaller’ attenuation is the one that delivers a higher probability of completing the outcome in any given time:

$$(\forall_x \Delta Q_1(x) \leq \Delta Q_2(x)) \equiv \Delta Q_1 \geq \Delta Q_2$$

This partial order has a ‘top’ element, which is simply perfect performance: $\top \equiv (\forall_x \Delta Q(x) = 1)$, and a ‘bottom’ element, which is total failure (an outcome that never occurs): $\perp \equiv (\forall_x \Delta Q(x) = 0)$.

Using this partial order, we can write specifications for system performance by requiring the delivered ΔQ to be less than or equal to a predefined bounding case. Where the delivered ΔQ is strictly less

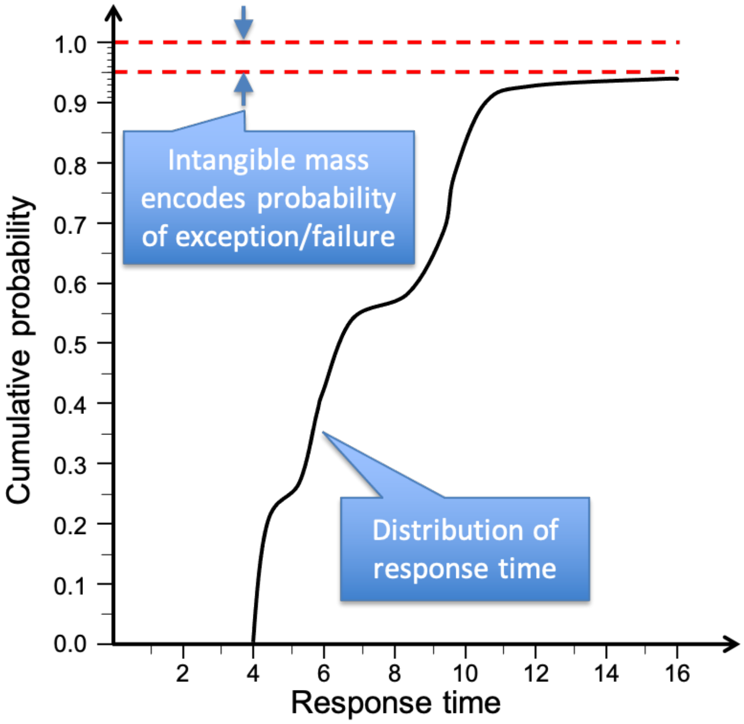


Figure 1: Cumulative distribution of an IRV

than the requirement, we say there is *slack*; when it is not less than or equal to the requirement, we say there is a *hazard*. More details are given in [9].

3.1 Application performance depends only on ΔQ

Applications depend on information to complete computations. To provide appropriately timely outcomes, this information needs to be delivered in a timely and correctly sequenced manner. If information takes too long to arrive (and/or too much of it is missing) then the computations cannot proceed, and the application fails to deliver the requested service with acceptable performance. An application's outcome will typically involve several information exchanges, some done sequentially, some done concurrently, that combine to enable a service. For a given outcome there will be bounds on the level of deviation from the ideal of instantaneous and loss-less transmission its information exchanges can tolerate.

Different components of a distributed application (e.g. a client and a server) exchange information as streams of packets. If those packets were all delivered instantaneously (i.e. if there were no attenuation in the translocation), and the computational components performed correctly, the application would be guaranteed to work. However, as discussed above, there will be some delay and occasionally packets may be lost, i.e. there is quality attenuation, ΔQ. Whether the application still delivers fit-for-purpose outcomes depends entirely on the extent of the quality attenuation (the 'magnitude' of ΔQ) and the application's sensitivity to it. The layering of network protocols isolates the application from any other aspect of the packet transport. This is such an important point it is worth repeating: the great achievement of network and protocol design has been to completely hide 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: ΔQ. Conversely, all that the network has to do to enable distributed applications to perform correctly is to deliver their packet flows with suitably bounded ΔQ. Establishing these bounds for particular applications is discussed in BBF MR-452.4 [13].

Bandwidth/capacity is necessary, but not sufficient. In fact, ‘insufficient bandwidth’ really means ‘at the desired offered load, the resulting packet loss/delay exceeds the acceptable performance bounds of the application’. Quality Attenuation unifies packet delay and loss (capturing them as statistical distributions) which in turn, also reflects network performance in terms of offered load compared to available bandwidth/capacity.

3.2 Decomposition of ΔQ

The ΔQ approach represents data transport quality as a set of distributions of delay with associated probabilities of loss. The measure of ΔQ for a given path can conveniently be broken into three components:

- $\Delta\mathrm{Q}_{|G}$ This is the distribution of inherent delay and probability of loss introduced by the path itself, which includes the time taken for signals to traverse it. It can be thought as the minimum time taken for a hypothetical zero-length packet to travel the path. In many cases this is effectively constant for relatively long periods of time, in which case it can be represented by a single delay value (for typical broadband networks, a convenient unit is ms). If characteristics of the path result in a baseline loss rate that is independent of packet size, this is included here, and $\Delta\mathrm{Q}_{|G}$ would then be represented as a pair (delay, loss probability);
- $\Delta\mathrm{Q}_{|S}$ This distribution is that part of ΔQ that is a function of packet size and incorporates effects like serialization and de-serialization time. $\Delta\mathrm{Q}_{|S}$ is a function from packet size to delay, which is usually monotonic and in many cases broadly linear, in which case we can represent it by a simple slope parameter, with the dimensions of time/length (for current network interface speeds, a convenient unit is μs/byte). If characteristics of the path result in a baseline loss rate that depends on packet size, for example due to a constant probability of corruption of each byte, this is included here and $\Delta\mathrm{Q}_{|S}$ could be represented as a pair (delay/length, probability/length);
- $\Delta\mathrm{Q}_{|V}$ This is the distribution of delay and loss introduced by the fact that the network is non-idle. This is modeled as a random variable, whose distribution may vary by time of day etc.. This can typically not be reduced to a single number, although moments of the distribution can be useful. The zeroth moment is the total probability, whose intangible mass represents loss; the first moment is the mean variable delay, measured in s; the second central moment is the variance, whose square root is the standard deviation, also measured in s. Loss that results from competition for shared finite resources such as interface packet buffering is included here (typically this is the main source of loss in modern networks).

In fixed-line networks, $\Delta\mathrm{Q}_{|G,S}$ tends to be quite stable, simply tracking routing changes, for example. Decomposing ΔQ in this way thus allows for a dramatic compression of the amount of data involved in continuously measuring the performance of a network path.

These measures are a superset of typical measures used today: for example, loss rate is encoded in the intangible mass of ΔQ, and jitter can be derived from $\Delta\mathrm{Q}_{|V}$.

The approach is compared against other application quality of experience measurement approaches in §4 of TR-452.1.

3.3 Composition and conservation of Quality Attenuation

Services are typically implemented in layers, in which an outcome at one layer is dependent on one or more outcomes at a lower level. This dependency translates into a relationship between the ΔQ of the outcome of interest and the ΔQs of the lower-level outcomes on which it depends. This relationship may be complex and nonlinear, but will typically be monotonic (in the presence of work-conservation), in that a larger ΔQ for one of the lower layer outcomes will imply a larger ΔQ for the higher layer one also. This is illustrated in figure 2, which shows how the time to complete a 1KB HTTP transfer depends on two particular aspects (mean delay and average loss rate) of the packet-level ΔQ. Where the behavior of the application/protocol is known, the relationship is calculable a priori.

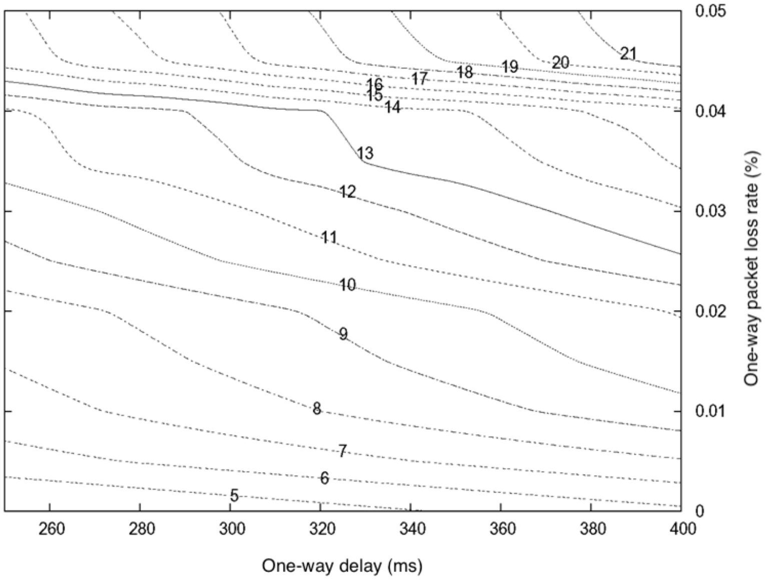
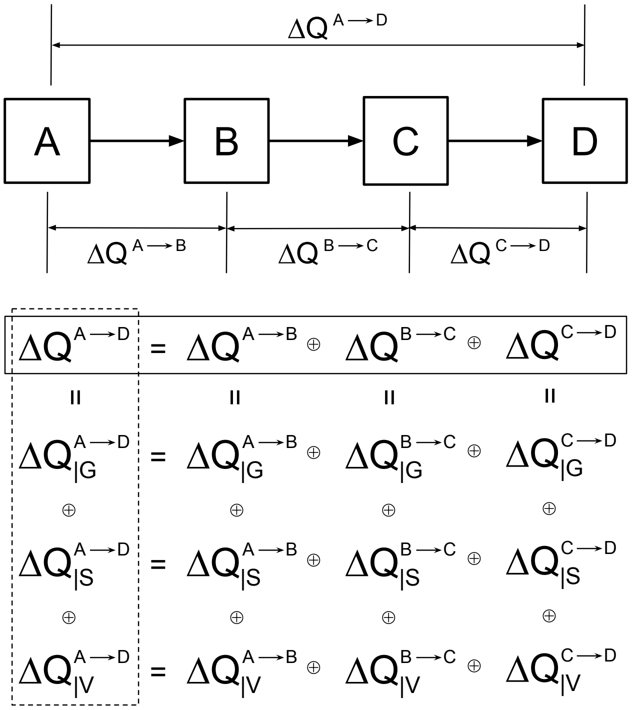


Figure 2: Time-to-complete contours of an HTTP transfer

Furthermore, quality attenuation is ‘additive’ within a single layer of a system. When an outcome depends on a sequence of steps (such as computations or forwarding of PDUs), the ΔQ of the whole outcome is the ‘sum’ of the ΔQs of the individual steps (if the ΔQs are independent IRVs the summation operation is simply convolution). This additivity distributes across the decomposition into components discussed in Section 3.2. This ‘compositionality’ is illustrated in Figure 3.

It is this compositionality that makes quality attenuation an effective measure for managing the performance of a large distributed system. It allows performance management to be devolved to subsystems and lower layers, by means of quality attenuation ‘budgets’. Compositionality and monotonicity guarantee that keeping lower-layer and subsystem attenuations smaller than their budgets ensures that the attenuation of the overall outcome is smaller than the requirement, which is to say that its performance target is met.

ΔQ is ‘conserved’ in the sense that any delay in delivering an outcome cannot be undone, nor can exceptions/failures/losses be reversed (at least not without incurring more delay). Thus, while different aspects of ΔQ can be traded to some degree, ΔQ as a whole cannot be reduced without changing the loading factor [6].



$$
\begin{array}{ccccccc}
\Delta Q^{A\to D} & = & \Delta Q^{A\to B} & \oplus & \Delta Q^{B\to C} & \oplus & \Delta Q^{C\to D} \\
\| & & \| & & \| & & \| \\
\Delta Q^{A\to D}_{|G} & = & \Delta Q^{A\to B}_{|G} & \oplus & \Delta Q^{B\to C}_{|G} & \oplus & \Delta Q^{C\to D}_{|G} \\
\oplus & & \oplus & & \oplus & & \oplus \\
\Delta Q^{A\to D}_{|S} & = & \Delta Q^{A\to B}_{|S} & \oplus & \Delta Q^{B\to C}_{|S} & \oplus & \Delta Q^{C\to D}_{|S} \\
\oplus & & \oplus & & \oplus & & \oplus \\
\Delta Q^{A\to D}_{|V} & = & \Delta Q^{A\to B}_{|V} & \oplus & \Delta Q^{B\to C}_{|V} & \oplus & \Delta Q^{C\to D}_{|V}
\end{array}
$$

Figure 3: Additivity of ΔQ by components.

4 Measuring and decomposing ΔQ

ΔQ is the distribution of delay/loss along a network path, which is measured by timing packets between the two or more points. This requires combining observations from different points, which typically raises complex issues of clock synchronization, and conflates issues of inherent delay, serialization time, etc.. However, by combining measurements of an outward and return path and processing the measurements as described below, these different factors can be teased apart.

Passage time of particular packets between two or more points are the raw material for measuring ΔQ. The packets observed can either be selected from a stream already transiting between the two points or belong to specific streams of test packets that follow the same path. Raw multi-point timing measurements produce a scatter plot of points characterized by *(time, size, delay)* tuples. Plotting measured packet delay versus measurement time instance reveals little structure. However, plotting measured packet delay versus packet size shows that for the same packet size, there is variability in the delay which is a function of instantaneous load on the various network elements. With a sufficient number of measurements, the minimum delay for each packet size in the scatter plot is attained when the variable delay due to contention is close to zero.

The basic processing steps for the estimation of ΔQ components from a set of measurements can be summarized as follows:

1. Arrange the population of measurement results by packet size to obtain an array of: *(pktID, pktsize, ΔQ(pktsize))*;
2. Construct the measurement sample population: *(pktID, pktsize, min ΔQ(pktsize))* ;
3. Use linear regression to fit a line through this population;

4. The slope of this line is an estimate of $\Delta Q_{|S}$;
5. The intersection point of the line with the pktsize $= 0$ axis is an estimate of $\Delta Q_{|G}$;
6. The distribution of the remaining ΔQ when $\Delta Q_{|G}$ and $\Delta Q_{|S}$ have been subtracted is an estimate of $\Delta Q_{|V}$.

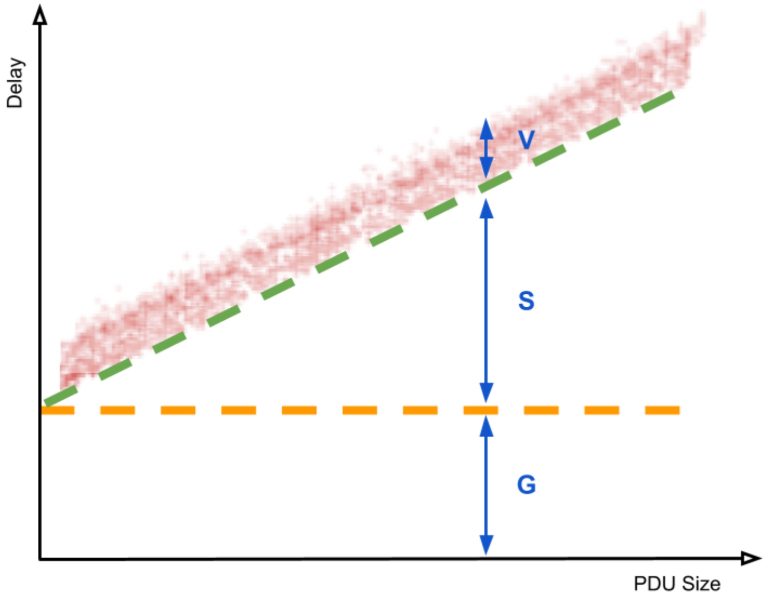


Figure 4: Extraction of ΔQ components

This is illustrated in figure 4.

This approach relies on two assumptions:

1. Larger packets take at least as long to transmit on average as smaller ones (i.e. $\Delta Q_{|S}$ is a monotonically increasing function of packet size);
2. At least some of the sample packets of any given size (bucket) experience negligible delay from contention for resources (i.e. the minimum of the set of samples from $\Delta Q_{|V}$ tends to zero as the sample size increases).

Provided both an outward and a return path are measured, clock skew between the two ends of the path can be compensated for, given some assumption regarding the $\Delta Q_{|G}$ (for instance, that it is symmetrical).

4.1 Using random sampling

The PASTA principle [11], and its more general case the Arrival Theorem [1], ensures that "probability of the state as seen by an outside random observer is the same as the probability of the state seen by an arriving customer" (in networking the 'customers' are packets). There are two ways to create a 'random observer': randomly sample existing traffic; or (relying on the arrival theorem) randomly insert packets that are tracked. Either approach ensures that the distribution of the attenuation captured by the samples tends to the distribution of the attenuation that the actual traffic experienced.

Typically, a traffic stream of randomly-sized packets with an overall rate of $16kbps$ is sufficient to measure the ΔQ along a network path. The PASTA principle ensures that any attenuation process, whether random or regular, is reflected in the distribution - irrespective of the duration of the attenuation. An added bonus is that any transient attenuation that lasts more than an inter-arrival time between sample packets (between 100ms and 500ms) is also captured in the observation time series. So, a queuing delay increase of, say 5ms, that persists for 500ms is immediately identifiable whereas an attenuation lasting between 1ms and 50ms is captured in the distribution, even when it occurs only infrequently.

There is an underlying trade-off: ‘time resolution’ and rate of convergence of the distribution can be improved by increased packet rate. This comes at the cost of an increase in the perturbation of measurements caused by the increased load from additional measurement packets. The optimal point varies between use-cases: pre-deployment testing of higher-rate equipment may require higher sample data rates, whereas wide scale deployments have to consider all the associated costs.

5 Applications of ΔQ measurements

ΔQ has been compared to measurement tools such as Real-time Response Under Load (RRUL). While it is true that RRUL and ΔQ can both be used to assess performance under maximum load, ΔQ has many other possible use-cases.

ΔQ has often been used for continuous measurement of a running system, such as end-to-end monitoring of the performance of a single application. This is practical due to the low overhead imposed by ongoing measurements. ΔQ can also be used to assess the performance of a single network component, or any sequence of components, under various loads ranging from nearly idle to overload. The dynamic response to changes in network load can also be accurately captured, which is useful for systems that alter their performance as a function of load: eg. power save mode, or alter their load as a result of increased attenuation: e.g. VoD systems that use adaptive bit-rate encoding.

In the longitudinal setting the ΔQ effects of load balancing as well as link / route failures (and recovery) can be explored. The compositional nature of ΔQ allows for identification of issues on common paths where there is an ensemble of measurements.

Conversely, A/B testing approaches have been used to evaluate architectural changes (e.g. quantify effectiveness of in-field technology updates), to compare the effective delivered quality from different geographical locations (useful for identifying MEC, cache and CDN placement), and to validate the equivalence of service over network infrastructure (e.g. IPv4 *v* IPv6 routing or service equivalence across vendor change).

ΔQ’s compositional properties allow for various forms of analysis and extrapolation. For example, it is possible to extract the ‘structural’ quality attenuation ($\Delta Q_{|G,S}$) from measurements and combine previous measurements to predict, for example, the effects of different path choices in a network architecture.

These properties also allow detailed measurements to be taken without excessive overheads. Let’s say we set up a series of measurement points along a network path where performance is important. To make sure we are notified in case the performance degrades, we set an alarm which triggers when the end-to-end ΔQ is above a certain threshold. For simplicity we assume that a single link on the path is responsible for the issue, but this is not a requirement in general.

The alarm triggers - now, what do we do next? We know that the end-to-end ΔQ is too large, but why? And how do we fix it?

The compositionality of ΔQ allows us to find the problem. Along the path, we have passive observation points at strategic locations. These observers see the measurement traffic go by, and store a local log of the last ≈ 60 seconds of observations. When the end-to-end alarm triggers, we can send a message to these observers and ask them for a copy of their log. We can start by asking the observer at the mid-point of our end-to-end link. This allows us to determine which side of the mid-point is to blame for the extra delay and packet loss. We can then ask more observation points until we know exactly where the problem occurred.

Using this approach minimizes the overhead incurred by continuous measurements while retaining the ability to pin-point the source of problems to whatever degree is economically feasible. Adding more observation points improves the granularity of the localization but does not increase the measurement overhead.

6 Conclusions

One of the complicating factors in Internet service delivery is that any statement such ‘ISP A is better than ISP B for X’ is highly contextual. It is technically feasible to deliver assured worst case quality attenuation to defined streams along a managed path, but the end-to-end performance is still vulnerable to (seemingly) minor changes in the wider operational environment (WiFi conditions, other traffic load at the end user premises, routing changes in the wider Internet) outside the direct operational control of an ISP.

The technical way forward on this may be to demonstrate where, along the end to end path, the quality attenuation is accruing with clear quantification as to where the dominant factors are located. This would provide, in the first instance, a way for the more engaged end-user to assure themselves of the veracity of any third party claims.

There are multiple stakeholders in any such conversation: regulators, marketing and legal departments as well as politicians. The intrinsic economic value and the potential effects of a miss-step mean that this is not a purely technical issue.

Figure 3 of MR-452.1 [12] illustrates how the ΔQ framework might be used in this sort of context.

We would suggest that some consideration should be given to the issues of how network measurements could be used in practice. During work for OfCom by some of the authors [8], the extent to which measurement and standardization can be used as ‘weapons’ in various settings became clear. This highlighted, to us, the need to consider how the underlying veracity of any measurement system could be evidenced and how its metrology would need to be carefully considered. As a first step, work on the standardization of conformance criteria for this measurement approach are underway in the Broadband Forum.

Given the substantial economic value in all aspects of the delivery chain, any measurement mechanism is likely to end up embedded as performance criteria in contractual relationships as well as regulatory expectations. Consideration needs to be given how any proposed measurement approach would fare under legal scrutiny.

References

[1] Søren Asmussen. Queueing networks and insensitivity. *Applied Probability and Queues*, pages 114–136, 2003.

[2] K. Hedayat, R. Krzanowski, A. Morton, K. Yum, and J. Babiarz. A two-way active measurement protocol (twamp). RFC 5357, RFC Editor, October 2008.

[3] Lucian Leahu. *Analysis and predictive modeling of the performance of the ATLAS TDAQ network*. PhD thesis, Bucharest, Tech. U., January 2013.

[4] G. Mirsky, G. Jun, H. Nydell, and R. Foote. Simple two-way active measurement protocol. RFC 8762, RFC Editor, March 2020.

[5] Jonathan Newton. Use of ΔQ to manage customer SLA. Marketing Report MR-452.2, Broadband Forum, July 2021.

[6] David C Reeve. *A New Blueprint for Network QoS*. PhD thesis, Computing Laboratory, University of Kent, Canterbury, Kent, UK, August 2003.

[7] David C Reeve, Neil J Davies, and Dale F Waldo. Constructing predictable applications for military ad-hoc wireless networks. In *MILCOM 2006-2006 IEEE Military Communications conference*, pages 1–7. IEEE, 2006.

[8] Predictable Network Solutions. Assessment of traffic management detection methods and tools. Technical Report MC-316, Ofcom, August 2015.

[9] P. Thompson and N. Davies. Towards a rina-based architecture for performance management of large-scale distributed systems. *Computers*, 2(53), June 2020.

[10] Peter Thompson and Rudy Hernadaz. Quality attenuation measurement architecture and requirements. Technical Report TR-452.1, Broadband Forum, September 2020.

[11] Ronald W Wolff. Poisson arrivals see time averages. *Operations research*, 30(2):223–231, 1982.

[12] Gavin Young. Motivation for quality verified broadband services. Marketing Report MR-452.1, Broadband Forum, October 2019.

[13] Gavin Young. QED uses in lab evaluation & network design. Marketing Report MR-452.4, Broadband Forum, Feburary 2021.