QoS
is meaningless for
Rich Communications Services

NFVRG

IETF-101

Yaakov (J) Stein

RAD

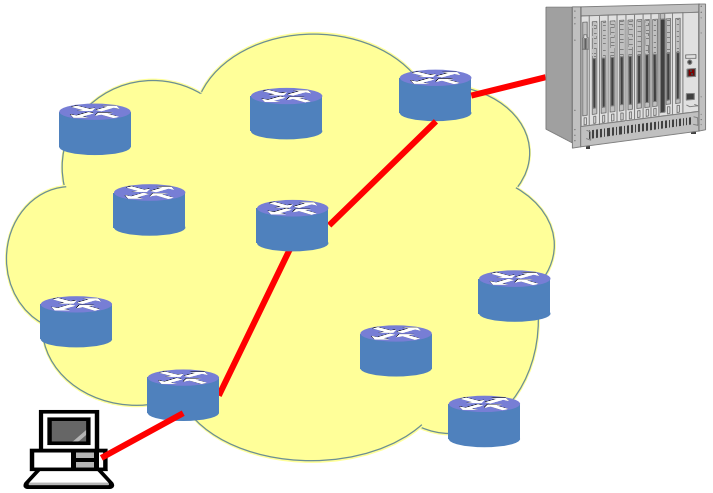
Communications services

A major paradigm major shift has occurred in networking spurred by NFV and in particular Distributed NFV

The traditional communications service was a *pure transport service*

Transport bits:

- from site X to site Y (or between N>2 sites)
- with data rate at least R
- with latency no more than L

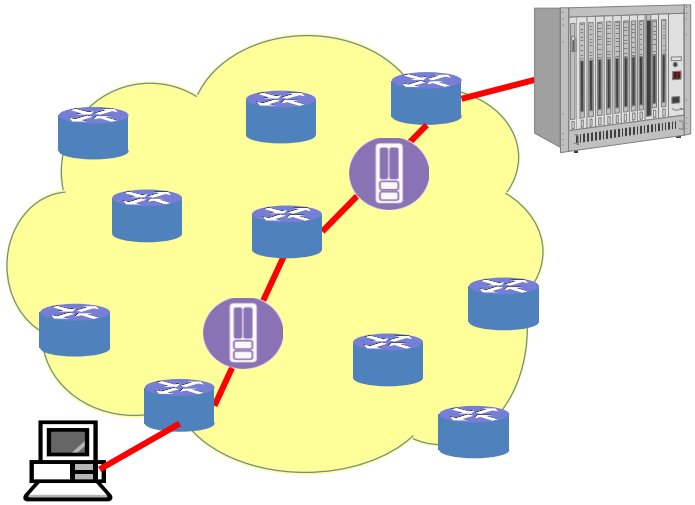


Rich communications services

Today, the service provided is much *richer*

Provide a rich service:

- from site X to site Y (or between N>2 sites)
- with application information rate at least R
- with experienced latency no more than L
- while performing (virtual) network functions A, B, C and D



QoS KPIs are proxies

Customers have grown accustomed
to receiving free or low flat-rate best effort service
and are only willing to pay for services
with QoE guarantees

There is a huge information base on how

- how to *guarantee* QoS KPIs
- to *measure* QoS KPIs

QoS KPIs are useful because

- they are straightforward to measure
- they perform well as *proxies* for subjective QoE

All current SLAs are based on QoS KPIs and not directly on QoE

QoE as a function of QoS

Many quantitative relationships between QoE and QoS have been found

QoE = f (application; QoSi)

justifying the measurement of QoS KPIs instead of directly estimating QoE

We know, for example:

- for conversational voice – how PLR/delay determine MOS given voice codec
- for streaming video – how PLR/PDV determine perceived quality given video codec
- for web browsing – how delay determines ApDex

QoS for rich services

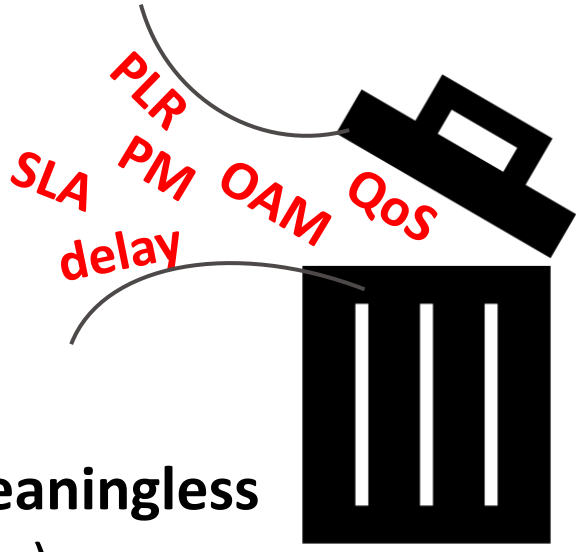
In fact, the *only* reason to guarantee/measure a QoS KPI is its relationship to QoE

but that relationship has only been established for *traditional* services

It turns out that one can **prove** that for rich communications services there is no such relationship

QoE ≠ f (application; QoS_i)

Hence **QoS and conventional SLAs are meaningless** (and NFV makes the situation even worse)



How can this be proven ?

The proof is based on
thought experiments (AKA *gedanken experiments*)

In each such thought experiment
- we pick a **KPI**

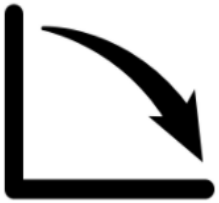
and
- show a **network function**
  that makes that KPI irrelevant to QoE

To demonstrate the principles involved
we'll show a few of these thought experiments

Packet loss can be *problematic*

Experiment 1 **Intrusion Protection Systems**

We are used to Packet Loss leading to QoE degradation
more specifically, increased PLR means decreased QoE



An IPS function **discards packets** that it deems to be malicious
thus **leading to an increased PLR**

Discarding these packets are in the user's best interest
thus the experienced QoE should increase

But there are much more convincing arguments!

Packet loss can be *meaningless*

Experiment 2 **TCP proxy**

A TCP proxy is placed near the middle of an end-to-end TCP session

The transmitted byte-stream is maintained
but its segmentation is not

For example, 3 packets may enter the proxy, and either 2 or 4 exit it !

Thus, **PLR can be high or even *negative* without affecting the QoE!**

We could abandon counting packets and measure *traffic volume*
(the *number of bytes* received irrespective of packetization)

So, let's check if *traffic volume loss* is a good QoS KPI

Volume loss can be *meaningless*

Experiment 3 **WAN optimization – compression**

Compression here can mean

- lossless data compression
- data deduplication
- audio or video compression, transcoding or transrating

All of these mechanisms

decrease the traffic volume without affecting QoE

So, traffic volume is not a relevant factor in determining QoE

The remedy is to completely abandon measuring byte volume and to measure *Shannon information* !

Information loss can be *meaningless*

Experiment 4 **WAN optimization – caching server (CDN)**

A caching server stores information
that may be consumed multiple times

When a flow contains cached information
zero information will transferred up to the cache
but the QoE remains unaffected

So, even measured Shannon information loss
can not be used as an end-to-end QoS parameter!

Note: Synthetic OAM packets aren't a fix

Network engineers will immediately object to our line of reasoning

Certainly PLR is well-defined
and the fault lies totally with our measurement methodology!

The proper way to measure PLR in such cases
is to introduce synthetic OAM packets
designed to bypass the computational functionality
and thus measure true end-to-end transport PLR!

That argument is completely true, and completely irrelevant !

We aren't interested in measuring QoS parameters as an academic exercise
The purpose of measuring them is to predict QoE on user traffic

Traffic that does not traverse all the elements of user packets
i.e., that is not *fate sharing* with true user traffic
can not be expected to assist in the prediction of the QoE of such user traffic!

Delay may be *meaningless*

The 2nd most useful QoS parameter
is end-to-end propagation delay

Of course, many of our previous examples
already cast doubt on the meaningfulness of delay

If packets are combined and re-segmented as in a TCP proxy
then we need to measure delay of individual bytes

If packet contents changes as in experiment 3 (compression)
then **byte delay becomes meaningless**

If packets are not even sent as in a caching server
then **propagation delay is undefined**

But, there is an even stronger argument !

Delay can be *unrelated to QoE*

Experiment 5 **Web browsing**

Studies show that users

- are usually satisfied if web pages stabilize in less than 2 seconds
- are usually frustrated
  if web pages don't stabilize within 8 seconds

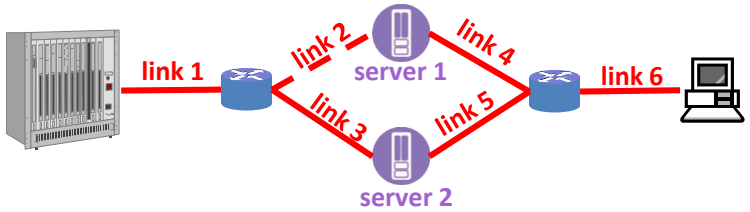
The browser is a software function that is part of the service and runs software (e.g., javascript) downloaded as data

This software may added an unbounded amount run-time before finally stabilizing the representation of the web page

Thus, delay from request to page stabilization is not uniquely determined by network delay!

Link failure can improve QoE!

Experiment 6 **Rerouting or protection switching**



A rich communications service

- initially traverses links 1, 2, 4, 6 and utilizes server 1

Due to a link 2 failure, the service is rerouted

- now traverses links 1, 3, 5, 6 and utilizes server 2

Server 2 happens to perform the functionality better

- due to upgraded software
- or more available CPU power and/or memory and/or storage

causing the QoE to improve!

Thus, **link failure may lead to QoE *improvement* !**

How does NFV affect this result?

One may be able to *work around* these results
for stationary network functionalities in known locations

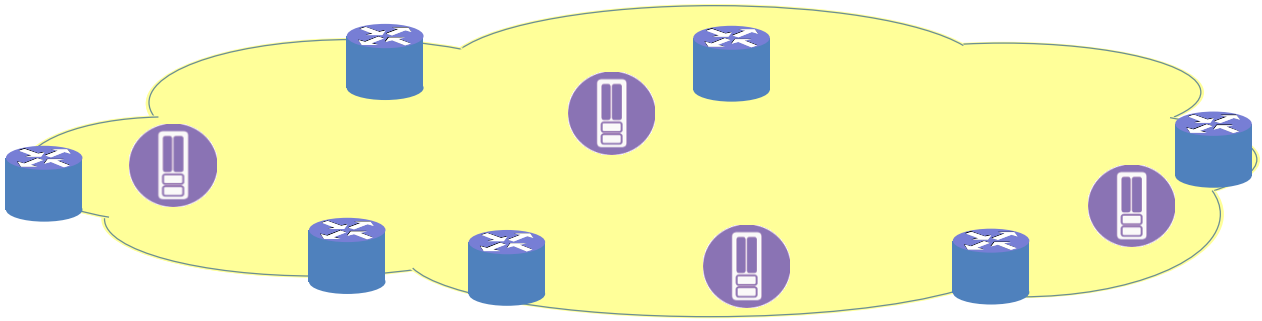
But NFV facilitates

- developing new functionalities
- dynamically inserting/moving/reconfiguring functionalities

so that

- we can not make assumptions on what functionalities do
- we can not make assumptions as to where functionalities are

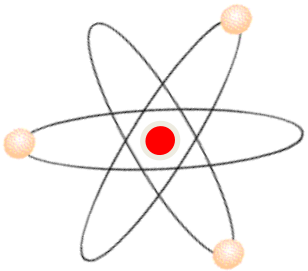
So, with NFV we must pessimistically assume
that any of the aforementioned problems may occur anywhere!



Possible solutions (WIP)

So, if QoS is meaningless, how can we estimate QoE ?

- directly access QoE estimation by application end-points
- indirectly estimate QoE based on user behavior
- measure QoE of synthetic but *fate-sharing* content
- correlate QoS KPIs with
  - NFVI KPIs (e.g., computation load, memory usage) *and*
  - VNF KPIs (as collected by VNFM)
- use machine learning techniques to predict QoE based directly on packet flows and NFV information without extracting traditional QoS KPIs



Thanks for listening !

comments appreciated

Yaakov_S@rad.com