# TIA Business Performance Community
**(formerly QuEST Forum)**

# Virtual Network Function Failure Frequency (VFF)
# Release 1.0

## NFV Strategic Initiative -
## Product Category
## Normalization Sub team

## Measurement Proposal

24 July 2018

## NFV Strategic Initiative
## Product Category Normalization Sub – Team

**Adtran**

Ed Bryan
Dave Schenkel

**AT&T**

Beth Ford
Brad Ruark
Taso
Devetzis

**CenturyLink**

Michael
Bugenhagen

**Cisco**

Tammy Gage
Tom Land

**Fujitsu**

Ashok Dandekar
Scott Jorrey
Masahiro Shimbashi

**Nokia**

John Wronka
Jose Marquez
Ben Jernigan

**Ribbon**

Paul Smith

**TIA**

Tom Yohe
Ken Koffman

**Telus**

Mike Newey

**Verizon Wireless**

Bryan Pollard

## 1.0 Introduction

This document describes a new quality measurement in support of Network Function Virtualization called the Virtual Network Function Failure Frequency (VFF).

This is an output of a sub-team of the TIA Business Performance Community NFV Strategic Initiative. As a brief background, the team has been looking at various measures that could be used to assess NFV quality, including ETSI NFV Service Quality Measures [1] and others. While numerous measures exist, and typically not documented in significant detail, the team developed the VFF concept as an overarching measure to assess the quality of Virtual Network Functions, and combines many attributes of the other measures investigated into one.

Simply put, VFF is calculated as the number of VNF failures per normalization unit. The team thoroughly studied two normalization units: VNF Instances and Virtual CPUs and recommended that both be considered. Other normalization units such as capacity could be explored, but are not addressed here.

Given the history of QuEST Forum (now known as the TIA Business Performance Community), and its ownership of the ICT TL 9000 Quality Management System, including the Measurements Handbook [2], the team took the approach of documenting the VFF in a TL 9000 measurement format. This format provides for each measurement a purpose, description, terminology, detailed rules for what failures are counted/excluded, along with calculations/formulas, discussion of data sources and examples.

Though documented in a TL 9000 format, it is not intended that the VFF become a TL 9000 measurement to be incorporated in the TL 9000 Measurements Handbook. It is more similar to network performance measures, but with a focus on VNF quality.

While this document describes the VFF, it does recognize many challenges that would benefit from future investigation:

- How are VNF failures best identified? For example, what role do orchestrators, VNF managers, play in capturing failures?
- If a VNF fails, how can failures be distinguished as caused by inherent quality issues, the NFV Infrastructure (NFVI), Management and Orchestration (MANO), or interoperability with other physical or virtual network functions?
- What are the best methods for capturing normalization units such as VNF instances or virtual CPUs? Are there other normalization units to be considered (e.g. capacity)?
- How could future products be designed to allow a standard, uniformly captured, segregated set of VFF measures to assess overall quality, including causes inherent to the VNF, NFVI, MANO and interoperability?

The NFV team appreciates any comments you may have on this document, and thoughts for future work. The team also hopes this work may motivate operators, suppliers and other standards organizations to build on this effort, intended to address the quality of VNFs as fundamental to Network Virtualization which is fundamental to the deployment of 5G networks.

## 1.1 Virtual Network Function (VNF) Failure Frequency (VFF)

### 1.1.1 General Description

The VFF measures the failure frequency of a Virtual Network Function (VNF). The number of failures may be normalized by VNF instances (VNFi) or Virtual CPUs (vCPUs).

VNFi as a normalization unit is the closest equivalent of the traditional normalization unit of Network Element (NE) for physical network functions. vCPUs are external resources provided to the VNF instances. While VNFi as a Normalization Unit appears to be closer to the traditional approach, vCPUs as Normalization Unit provides results that are more vendor neutral and considers the potentially different resources required by VNFs from different suppliers.

Consideration could also be given to normalize by capacity, but for the purposes here the focus is on VNFis and vCPUs.

### 1.1.2 Purpose

The purpose of the VFF is to provide insights into VNF failures given their important role in replacing traditional network functions (NFs) that are often deployed as network elements (NEs) comprised of proprietary hardware and/or software.

The VFF guides organizations to monitor VNF quality and effectively manage the costs required to maintain and service the VNF product. Due to product configuration, network resiliency and VNF redundancy and scalability, failures reported in the VFF measurement do not necessarily cause service loss to an end user. However, each failure results in a maintenance action (MANO driven or manual) and thereby impacts operation and/or maintenance costs.

VNF failures can have many cause categories:

1. Inherent quality, e.g. defects within the VNF software itself which may also be exhibited in associated containers, virtual machines (VMs), and Virtual Network Function Components (VNFCs).
2. Network Function Virtualization Infrastructure (NFVI) issues that may provide the VNF with insufficient compute, storage, or other resources such as accelerators and other components
3. Orchestration issues - inability of the orchestration platform to instantiate service based on service policy defined by the service (e.g. insufficient number of VNFs instantiated, VNF not instantiated with declared policy (e.g. processor or geographic affinity, VM flavor, VM requirements, VNF associated VMs that are dead on arrival due to NFVI or MANO issues).
4. Interoperability issues including interoperability with other NFs and VNFs.

Ideally the VFF should include failures for all above causes. For example, a major increase in the VFF could signal major problems in compute or storage resources that would need to be rectified before end-user impacts are observed.

In addition to capturing all causes, it would be beneficial to have a VFF for the various sub causes.

Unfortunately, capturing VNF failures is a challenge as mentioned in section 1.1.5, and being able to

classify them by various causes adds further complexities.

Thus, for the purposes here, the focus will be on item 1. above, inherent VNF quality as a start. The rationale for this as a start is that it may be easier to capture inherent failures than for the other causes. Also, it is independent of NFVI, orchestration and interoperability that can vary from operator to operator, or even network to network within an operator. Assuming one of the usages of the VFF may be comparison of one vendor's VNF with another vendors VNF, eliminating these other variables such as NFVI may provide a more suitable comparison.

### 1.1.3 Applicable Product Categories

The VFF can be applied to any VNFs where VNFis or vCPUs are suitable normalization units.

### 1.1.4 Detailed Description

#### a) Terminology

1. Virtual Network Function – [3] defines a VNF as "An implementation of an NF (network function) that can be deployed on a Network Function Infrastructure (NFVI)." For the purposes defined here more detail is added - a VNF is a software defined function aligned to telecom or network services and is suitable to be deployed over NFVI/Cloud infrastructure. A VNF may contain one or more VNF Components (VNFCs) to be deployed over one or more Virtual Machines, containers, or other virtualized components, and each virtual component may be provisioned with one or more vCPUs. The VNF can dynamically scale up or scale down by instantiating up/down VNFCs within the VNF, and/or NFVI resources (vCPUs, RAM, VMs) associated with the VNFCs and VNF.
2. VNF Instance - A running instance of the VNF including: run-time instantiation of the VNF software, resulting from completing the instantiation of its components and of the connectivity between them, using the VNF deployment and operational information captured in the VNF descriptor, as well as additional run-time instance-specific information and constraints [3]
3. VNF Component - internal component of a VNF providing a VNF Provider a defined sub-set of that VNF's functionality, with the main characteristic that a single instance of this component maps 1:1 against a single Virtualisation Container [3]
4. vCPU - vCPU is a virtualized CPU provided by the Hypervisor to a Virtual Machine, typically corresponding to the physical equivalent of a hardware CPU core, but vCPUs may be oversubscribed meaning there may be many more vCPUs than HW CPU cores in a server where each vCPU receives only a fraction of the capacity of a HW CPU. In the best possible scenario, a vCPU may be able to provide computing capacity equivalent of one physical core of the physical processor device, but not beyond.
5. Container- A container, in this document, refers to a generic atomic virtualization instance in OS-level virtualized environments (including jails, VEs, etc.).
6. OS-level virtualization **-** Operating-system-level virtualization refers to capabilities implemented at the OS-level that provide isolated execution environments for applications. Instances running under this regime (often referred to as containers) are afforded isolated resource views (processors, hardware, file systems, networks, etc.) that are separate from other containers, with the OS managing the name spaces, communication, and resource access. Containers typically encompass one or more processes implementing a VNF or part of a VNF.
7. Virtual Machine - virtualized computation environment that behaves very much like a physical computer/server.

NOTE: A VM has all its ingredients (processor, memory/storage, interfaces/ports) of a physical computer/server and is generated by a Hypervisor, which partitions the underlying physical resources and allocates them to VMs. Virtual Machines are capable of hosting a VNF Component (VNFC). [3]

8. Network Function - functional block within a network infrastructure that has well-defined external interfaces and well-defined functional behavior

   NOTE: In practical terms, a Network Function is today often a network node or physical appliance. [3]

9. Network Element - A system device, entity or node including all relevant hardware and/or software components. The Network Element (NE) must include all components required to perform the primary function of its applicable product category. If multiple FRUs, devices, and/or software components are needed for the NE to provide its product category's primary function, then none of these individual components can be considered an NE by themselves. The total collection of all these components is considered a single NE. [2]
10. Network Functions Virtualisation Management and Orchestration (NFV-MANO): functions collectively provided by NFVO (Orchestrator), VNFM (Virtual Network Function Manager), and VIM (Virtual Infrastructure Manager). [3]

## b) Counting Rules

1. Failures in category 1 in VFF description - inherent quality, e.g. defects in the VNF software itself which may also be exhibited in associated containers, virtual machines (VMs), and VNFCs
2. A failure shall be counted against the VNF if no failure cause can be determined.
3. For new VNFs, their releases, and/or updates being deployed, failures shall be counted starting with the first live deployment of the new VNF. In cases where a small percentage of VNFCs that comprise a VNF carry live traffic, all inherent quality VNF failures shall be counted.
4. If there is disagreement, doubt, or if the failure is due to multiple causes, the determination of whether the failure is inherent to the VNF itself (i.e. Category 1) the final determination shall be made by the customer.
5. If a VNF fails to initialize due to an inherent defect within itself, it shall be counted as a VNF failure.
6. If a VNF fails to terminate due to an inherent defect within itself, it shall be counted as a VNF failure.

## c) Counting Rule Exclusions

1. Failures are counted for a VNF only when the failure is within and inherent to the VNF itself. Failures caused by other products or conditions in the network are excluded. For example:
   a. NFVI issues that may provide the VNF with insufficient compute and storage resources or other resources such as accelerators and other components
   b. Orchestration issues - inability of the orchestration platform to instantiate service based on service policy defined by the service (e.g. insufficient number of VNFs instantiated, VNF not instantiated to the right policy, VNF associated VMs that are dead on arrival due to orchestration)
   c. interoperability issues including interoperability with other NFs and VNFs.
2. If, as a matter of policy, a customer does not make VNF failure data available to the organization, then the normalization units (VNFis, vCPUs.) deployed by that customer shall be excluded from the VFF measurements.
3. Failures that occur in labs or other trials that do not involve live traffic are not counted.
4. A failure shall not be counted if it is caused by a problem for which there is a fix available at no cost,
   a. And a decision has been made not to deploy the fix,

b. Or a failure occurs six months after the date the fix is generally available or other mutually agreed upon deployment period.

### c) Calculations and Formula

**Table 1.1-1 VFF Notation and Calculations**

| Identifier | Definition | Formula or Note |
|---|---|---|
| Afactor | Number of calculation periods in a year | Afactor=12 |
| *m* | Current calculation month | |
| $T_m$ | Number of total VNF Failures in month *m* | For the purposes here, we will be focusing on category 1 failures (e.g. inherent to the VNF). However, if all causes of failure could be identified, or sub-causes identified the calculation could be modified accordingly to include all or specific sub-cause(s) categories. For example, Category 2 failures due to NFVI could be counted as a separate category and would not be considered an exclusion. |
| $T_{vcpu}$ | Number of total vCPUs in operation at the end of month m | This may be an estimate. Typically, an average number of vCPUs are required to support a VNF instance. If this average number is known then the total number of vCPUs can be calculated as the product of Tvnfi * average number of vCPUs per VNF instance. |
| $T_{vnfi}$ | Number of total VNF instances in operation at month m | |
| $VFF_{vnfi}$ | VNF Failure Frequency using VNFis as a normalization unit expressed as failures/VNF Instance/year | $Afactor*T_m / T_{vnfi}$ |
| $VFF_{vcpu}$ | VNF Failure Frequency using vCPUs as a normalization unit expressed as failures/vCPU/year | $Afactor*T_m / T_{vcpus}$ |

## 1.1.5 Sources of Data

For traditional TL 9000 measurements there are methods for suppliers of physical products to estimate normalization units (e.g. NEs via shipping records) and in capturing required numerators (e.g. problem reports and outages through customer care systems).

In comparison, data sources and collection for the VFF calculations by necessity will require more inputs from operators and present additional challenges.

vCPU and VNFi normalization units may be captured using OpenStack scripts.

Failures present more of a challenge, especially if attempting to classify them into categories 1 to 4 mentioned previously in the "Purpose" section.

Various approaches could be used:

Virtual Infrastructure Management (VIM) (OpenStack) detection – bottoms up approach to monitor events for all VMs corresponding to VNFs. However, with this approach it will be difficult to distinguish failures inherent to the VNF (category 1) from those caused by the NFVI (category 2).

NFVO and VNFM based detection – top down approach to monitor VNFs at the orchestrator and VNF manager levels. The challenge with this approach is that it could be very product specific and dependent on the orchestrator or manager used.

A hybrid approach using both bottoms up and top downs is conceivable, but NFVO and VNFM products must meet some generic specifications that would be independent of the product used.

Finally, if TL 9000 Software Problem reports are being collected from the field for a VNF release, these could be considered as category 1 failures. While there are Critical, Major and minor problem reports, the recommendation would be to include only the critical and major problem reports.

### 1.1.6 Example Calculations

**Case 1 Using VNF Instance as Normalization Unit**

For a VNF product (e.g. session border controller), an operator has 10 VNF instances operating at the end of a month. These could be operating at different locations, and during the month 2 failures occur across these 10 VNFs. The operator now wants to estimate $\mathbf{VFF}_{vnfi}$ for the given month.

In this case, $T_m = 2$, $T_{vnfi} = 10$, and the resulting $\mathbf{VFF}_{vnfi} = 12*2/10 = 2.4$ failures/VNF Instance/year

**Case 2 Using vCPUs as Normalization Unit**

For a VNF product (e.g. session border controller), there are 10 VNF instances supported by 50 vCPUs operating at the end of the reporting month. (note that the 50 vCPUs is estimated knowing that on average 5 vCPUs support each VNF instance (10 instances * 5 vCPUs/instance = 50 vCPUs)). These VNF instances and vCPUs could be operating at different locations, but during the month 2 failures occur across these 10 VNFs and 50 vCPUs. The operator now wants to estimate $\mathbf{VFF}_{vcpu}$ for the given month.

In this case, $T_m = 2$, $T_{vcpu} = 50$, and the resulting $\mathbf{VFF}_{vcpu} = 12*2/50 = .48$ failures/vCPU/year

**REFERENCES**

[1] ETSI GS NFV-INF 010 V1.1.1 (2014-12) Network Functions Virtualisation (NFV);
Service Quality Metrics
http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/010/01.01.01_60/gs_NFV-INF010v010101p.pdf

[2] Quest Forum, "TL 9000 Measurements Handbook", release 5.0, July 2012 + Addendum Release 5.5 June 2017
http://www.tl9000.org/handbooks/measurements_handbook.html.

[3] ETSI NVF - ETSI GS NFV 003 V1.3.1 (2018-01) Network Functions Virtualisation (NFV);
Terminology for Main Concepts in NFV
http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_NFV003v010201p.pdf