University of Minho
School of Engineering
Department of Informatics

Joana Lourenço Carreira

Network and Cross-Platform SaaS Performance: a Case Study

February, 2020

University of Minho
School of Engineering
Department of Informatics

Joana Lourenço Carreira

Network and Cross-Platform SaaS Performance: a Case Study

Master dissertation
Master of Science in Network Engineering and Telematic Services

Dissertation supervised by
Professor Maria Solange Pires Ferreira Rito Lima

February, 2020

COPYRIGHTS AND TERMS OF USE BY THIRD PARTIES

This academic work may be used by third parties as long as internationally accepted rules and good practices are respected, specifically in terms of copyrights and all associated rights.

Thereby, this work may solely be used under the terms of the license presented below.

In case, the use of this work is required, under conditions not covered by the license presented below, a permission shall be obtained and the request directed to the author via the RepositóriUM of the University of Minho.

Attribution-NonCommercial-NoDerivatives
CC BY-NC-ND

ACKNOWLEDGMENTS

> *"When you get into a tight place, and everything goes against you till it seems as if you couldn't hold on a minute longer, never give up then, for that's just the place and time that the tide'll turn."*
>
> ***Harriet Beecher Stowe***

I would like to dedicate this work to all those who stayed by my side, choosing to believe in me even when I did not believe in myself; during the rocky process of analyzing massive data sets and writing this thesis, that at times seemed painfully impossible due to the inextricability of personal and professional life.

Thank you, mum and dad, for your great advice and for being there for me all the time.

A really great and warm thank you, to my exceptionally kind and beautiful seven-years-old daughter Inês. To you, my sweet girl, thank you for your patience while I had none to give you; for your altruistic attitude of sharing me with an attention seeking computer, while maintaining those sparkly eyes believing that better days would be coming. Thank you for those gestures that, though perceived as small, were the ones that made the greatest difference by showing me what really matters – *You have to keep trying, ladybug!*

To my baby boy, João, thank you for your talking eyes and contagious smile every morning. You made my days less somber and constant flow of positive energy.

I would like to thank as well to my significant other, Marek, for the time and infinite patience needed against my being so feisty when brainstorming and organizing my thoughts. Also, a special thanks for revising infinite times my grammar and spelling and for pushing me to "think outside of the box".

Last but not least, a special thanks to my teacher, Professor Solange Lima, for showing always availability and constant support, for being extra patient through all this rocky personal and academical intertwined and inextricable process, while at the same time making sure that the defined academic goals are achieved. I will be always grateful, as well, for your savviness and guidance during this work in moments that I could not see a path forth. Thank you for those breakthrough moments that allowed me to grow and finish successfully this work.

STATEMENT OF INTEGRITY

I hereby declare that I have conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Abstract

The traditional role of a personal computer is dramatically changing with the shift towards cloud-based services. Cloud computing and storage provides end-users with universal access to their data across various devices, coherent application and service experience, and substantially decreases hardware requirements for end-user clients (personal computers). However, this cloud-oriented paradigm requires the redesign of applications and services, as well as a serious analysis of the current situation and proper scaling of access networks since this new paradigm changes everyday habits of end-users.

This work is focused on the impact of cloud-based applications (using the paradigm of Software as a Service (SaaS)) on access networks, analyzes the network and the application behavior, while also addressing application usability and Quality of Experience (QoE) in different scenarios. A detailed study of the impact on access networks imposed by such cloud-based services (and vice versa) is currently missing, especially in the case of bandwidth-constrained, high-latency mobile access networks.

Furthermore, this work involves analysis of various cloud-based applications, namely office tasks (text, presentation, and spreadsheet editing), in different combinations and executed on different hardware and software platforms with different levels of integration with cloud-based services. Network traffic analysis will be executed, including collecting Wireshark traces of the generated and received traffic, correlated with specific executed tasks. The impact of network congestion and latency is also examined in the QoE-focused section. The work discussion is broken down into individual hypotheses, reflecting expectations regarding behavior of SaaS applications, data volume of the network, and QoE of the end user. Different end-user experience metrics are used in combination with network-based monitoring (including peak and average bandwidth measurements, latency, packet loss, etc.).

Keywords: Cloud computing, Google Docs suite, Microsoft Online suite, Network behavior, Quality of Experience, QUIC, SaaS performance, Traffic profile.

Resumo

O conceito tradicional de computador pessoal está a mudar drasticamente com a mudança de paradigma para a utilização de serviços disponíveis através da nuvem (*cloud-based services*). A computação e armazenamento na nuvem possibilita aos utilizadores finais o acesso universal aos seus dados através de diversos dispositivos, um acesso coerente às aplicações e, respetiva, qualidade de serviço, reduzindo substancialmente os requisitos de *hardware* dos utilizadores finais (nos computadores pessoais). No entanto, a mudança para o paradigma da computação em nuvem exige que se repense não só as aplicações e serviços, mas também exige um estudo sério sobre o panorama atual e a adequada escalabilidade das redes de acesso, uma vez que existe alteração nos hábitos diários dos utilizadores.

Este trabalho foca-se no estudo do impacto da utilização das aplicações da nuvem (usando o paradigma de *Software-as-a-Service* (SaaS)) nas redes de acesso, na usabilidade das aplicações e na Qualidade da Experiência (QoE) em vários cenários. Atualmente não existe um estudo detalhado sobre o impacto que estes serviços disponibilizados pela nuvem têm nas redes de acesso, especialmente nos casos das redes de acesso móveis que apresentam por si só restrições consideráveis de largura de banda e elevada latência.

O estudo contempla a análise da utilização de várias aplicações da nuvem, nomeadamente, tarefas de escritório (edição de texto, edição de apresentações multimédia, edição folhas de cálculo), combinadas e executadas em diferentes plataformas de *hardware* e *software* com diferentes níveis de integração com os serviços disponíveis na nuvem. Será efetuada uma análise do tráfego da rede, com recolha de *traces* do Wireshark do tráfego gerado e recebido, correlacionando-se com tarefas específicas. O impacto da congestão da rede e a latência são também examinadas na seção focada na QoE. A discussão do trabalho encontra-se distribuída individualmente pelas várias hipóteses formuladas e que refletem as expectativas relativamente ao comportamento das aplicações SaaS, volume de dados na rede e QoE por parte do utilizador. Diferentes métricas para a QoE são usadas, combinadas com a monitorização de parâmetros relevantes da rede (medições da largura de banda média e valores de pico, latência e perda de pacotes, etc.).

Palavras-chave: Comportamento da rede, Computação em nuvem, Desempenho SaaS, QUIC, Perfil de tráfego, Qualidade de Experiência, Suite da Google Docs, Suite da Microsoft Online.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ARP	Address Resolution Protocol
BR	Browser
DNS	Domain Name System
DHCP	Dynamic Host Configuration Protocol
FF	Mozilla Firefox
FTTC	Fiber to the Curb
FTTH	Fiber to the Home
FTTx	Fiber To The X (Home, Business, Curb)
GC	Google Chrome
GD	Google Docs application
GS	Google Slides application
GUI	Graphical User Interface
GX	Google Sheets application
IaaS	Infrastruture-as-a-Service
ICMP	Internet Control Message Protocol
IE	Internet Explorer
IP	Internet Protocol
ISP	Internet Service Provider/Internet Carrier
IT	Information Technology
KU	Kubuntu operating system
LAN	Local Area Network
MU	Multiple User
NAT	Network Address Translation (with Port Address Translation)
OS	Operating System
PaaS	Platform-as-a-Service
PAN	Personal Area Network
PC	Personal Computer
PO	Microsoft PowerPoint Online application
QoE	Quality of Experience
QoS	Quality of Service

RTT	Round Trip Time
SaaS	Software-as-a-Service
SaaS host(s)	Network of servers hosting the SaaS
SLA	Service Level Agreement
W7	Windows 7 operating system
W10	Windows 10 operating system
WO	Microsoft Word Online application
xDSL	Digital Subscriber Line, different versions
XO	Microsoft Excel Online application

1

INTRODUCTION

In an era of evolving society focused on rapid technological innovation, individual contributors and work teams require application platforms that inherently support efficient collaborative work while providing project management tools, and aid development of projects in a professional and timely manner.

In this context, cloud computing, storage, and application services are increasingly becoming part of everyday professional life of an individual. Both companies and individuals move to cloud-based solutions due to a perceived cost reduction, decreased application deployment and maintenance costs, and reduced security exposure, where updates, security patches, and everyday maintenance of such platforms are shifted to the cloud services' provider. A wide variety of cloud-based solutions have come into existence to accommodate the needs of individuals and organizations of various sizes. Such solutions include: a) access to infrastructure resources (i.e., Infrastructure-as-a-Service – IaaS) based on virtualization of physical resources; b) access to specific platforms while not having to worry with hardware specifications and only installing and configuring a specific software or tool (Platform-as-a-Service – PaaS); and c) access to specific applications (Software-as-a-Service – SaaS) without actually having to install them and, in some cases - purchase and license them as well. Cloud-based solutions offer therefore service flexibility without the upfront investment into the hardware and/or software infrastructure.

Software-as-a-Service (SaaS) has quickly achieved significant penetration in the market, with whole operating systems and associated hardware (for example, Google Chromebooks) built around the model of always-connected, cloud-based applications, where user data is stored in the cloud and it is downloaded only when and if needed. The increasing popularity of such cloud-based services brings questions about the impact of such solutions on access networks, especially in the case of bandwidth-constrained and high latency mobile and satellite-based networks. There are also questions associated with the promise of SaaS providing a fully cross-platform experience, where a simple Internet browser provides access to all and any application(s) a person might need. The Quality of Experience (QoE) for such solutions remains largely unexplored, representing one of the more interesting areas of the study undertaken in this work.

1.1 Motivation and Goals

Due to the increasing pervasiveness of SaaS applications and their popularity among both individual and corporate customers, a study comparing the behavior of both SaaS solutions provided by Google and Microsoft, assessing their interaction with access networks, and end user experience is needed. Describing individual SaaS performance in terms of a network profile allows for the better understanding of the suitability of such applications for several types of access networks. It also allows the end users to assess the current Service Level Agreement (SLA) they have with their Internet Service provider (ISP) and evaluate if the available Internet connection is sufficient to support the use of SaaS applications.

Therefore, the main goal of this work is to assess the performance of the network and of the cloud-based computing model (SaaS) for different access network types, while using different SaaS applications. To accomplish the main goal of this work, several more specific and intermediate goals were defined, as follows:

- analyze the behavior of different SaaS applications provided by Google and Microsoft to perform office tasks such as text, presentation, and spreadsheet editing;
- compare the behavior of individual SaaS applications in different network and end-client platform scenarios;
- examine correlation between individual SaaS applications and resulting network activity patterns under different browsers, operating systems, and network configurations;
- analyze transport protocols involved, traffic statistics (for example, traffic volume in upstream and downstream directions), and traffic profiles;
- identify tasks that are more network-intensive for each chosen SaaS application, as well as identify more network-intensive SaaS application types;
- verify the consistency of the behavior for the chosen SaaS application (text editing) for a combination of browsers and operating systems under different network configurations;
- and, finally, verify suitability of specific SaaS applications for the use with mobile access networks and identify possible constraints that may impact the quality of experience (QoE) of the end user, for the chosen SaaS application and both providers.

1.2 Work contribution

This work main contribution is focused on examining the interaction between Google/Microsoft SaaS and the access network, specifically, focusing on SaaS performance from the end-user perspective.

Taking the promise of cross-platform nature of SaaS and variety of access network scenarios possible into consideration, different hypotheses are presented and examined in detail addressing different aspects of SaaS operation. In particular, correlation between specific events and resulting network activity, independence of the specific SaaS from different platforms, types of network and transport layer protocols being used, packet size distribution and network load direction associated with specific SaaS, are all aspects covered in the scope of this work.

At last, the impact of network latency and packet loss is also examined in detail, with a proposal for definition of SaaS performance degradation severity levels. These performance degradation severity levels allow end users to grade SaaS performance in a more objective manner, while not requiring the network-level analysis. A clear correlation between increasing SaaS performance degradation severity level and network-level packet latency and packet loss is demonstrated, using high-latency and lossy mobile network environment for this purpose.

1.3 Dissertation layout

This dissertation is organized in five main chapters, starting with this Introduction.

Chapter 2 provides an introduction to basic concepts pertaining to cloud computing and additional information concerning related work where past and current developments and challenges in the field of cloud computing systems, and more specifically concerning the use of Software-as-a-Service (SaaS) solutions are described and literature referenced.

Chapter 3 describes the scope of this thesis, where individual hypotheses are formulated and presented, followed by the description of the network configuration used, the platforms and tools chosen to accomplish the goals defined in the Section 1.1. A description of how the scenarios are named for the sake of consistency and clarity through the document and a test taxonomy is also presented with a brief summary of the goals that are associated with each scenario. Finally, the last section of the Chapter 3 describes the methodology used to process and perform the analysis of the collected packet traces.

Chapter 4 is the core chapter of this work, where individual hypotheses are explained in detail with description of the expectations. Each hypothesis section comprises not only the obtained results but as well their analysis and discussion. Due to the extent of the work, some of them may present the full set of data to explain the reasoning behind the process, others may present one example explaining the similarities, and others may present only the most important cases or differences.

Chapter 5 presents the relevant conclusions of this work and the proposed future work, based mostly on unanswered questions that may be identified during or after the development and testing of individual hypotheses, especially when inconsistent or inconclusive results are observed.

2

State of the Art

This chapter consists of an overview of the cloud computing paradigm supported by the available literature to the present date. It presents and describes the concept of cloud computing, its base services models, advantages, risks and vulnerabilities and as well as the trade-offs that users face while providing a broader view of the prevalence of this model. SaaS-related work is also discussed, followed by identified open issues and challenges in this area.

2.1 Cloud Computing

According to NIST [1], "*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*". ISO/IEC [2] presents a similar definition, i.e., "*Cloud computing is a paradigm for enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning and administration on-demand.*". Both standards development organization define cloud computing in a similar way and describe enormous advantages of the use of this computing paradigm.

This new shift towards cloud computing has enjoyed enormous development and implementation investment over the last decade, as described in [3][4], having effectively become a widespread and readily available commodity service for any person with Internet access anywhere around the world.

The increasing penetration of high-capacity access networks and ever-growing demand for reliable storage drive the adoption of distributed and reliable cloud-based storage. Also, the inherent security risks and costs associated with managing applications on local machines drive the adoption of cloud-based applications for both individual as well as enterprise users.

Finally, the need for on-demand hardware (computer, processing, storage, etc.) drives the need for cloud-based infrastructures. This approach allows end users to save money, shifting the investment into cloud service providers' side, since they do not need to invest into dedicated IT infrastructures/hardware (traditional model) maintaining on-demand access to both hardware and software, tailored to their current demand. Additionally, the load balancing techniques employed by cloud service providers allow for almost infinite resource scaling; background data backups that provide much improved reliability, and the virtualization allows users to run any operating system and any set of applications without having to be

tied to specific hardware and software solutions. At last, the end users benefit, not only from the described reduced investment in resources, but they no longer need local IT support. They enjoy more limited security exposure due to unpatched vulnerabilities in client software and are able to access specifically required resources.

The cloud computing model has several advantages, namely improved fault tolerance and rapid disaster recovery for any critical applications and functions, simpler service deployment model, persistent accessibility from anywhere in the world using existing Internet transport architecture, improved and centralized application security for data and applications, and others. The ability to shift the storage and the computational loads between different data centers is especially critical for mission-critical applications that may not be affected by network performance degradation events. For example, a compute cluster can be moved from one end of the continent to another in a transparent manner to the end user, allowing a specific task to run uninterrupted.

2.2 Service models

Cloud computing is based on service models, with the following primary delivery models [1]:

- Infrastructure-as–a-Service (IaaS), providing access to fundamental resources, such as virtual machines and hardware resources, software and application deployment and configuration, virtual storage, etc.
- Platform-as-a-Service (PaaS), providing the runtime environment for applications, development and deployment tools, etc.
- Software-as-a-Service (SaaS), allowing to use software applications as a service to end-users, with limited configuration.

Individual service models can be presented graphically as shown in Figure 1.

IaaS and PaaS models are primarily directed at enterprise customers, looking for virtualized resources (physical machines, virtual machines, virtual storage) or runtime environment, especially for short periods of time, when specific resources can be made available on demand, without having to maintain hardware and software infrastructures within the given organization and keep the respective IT staff on payroll. This model improves the response time for most enterprises, allowing them to access compute and storage resources as needed, paying only for the consumed compute and storage, and releasing such allocated resources when they are no longer needed.

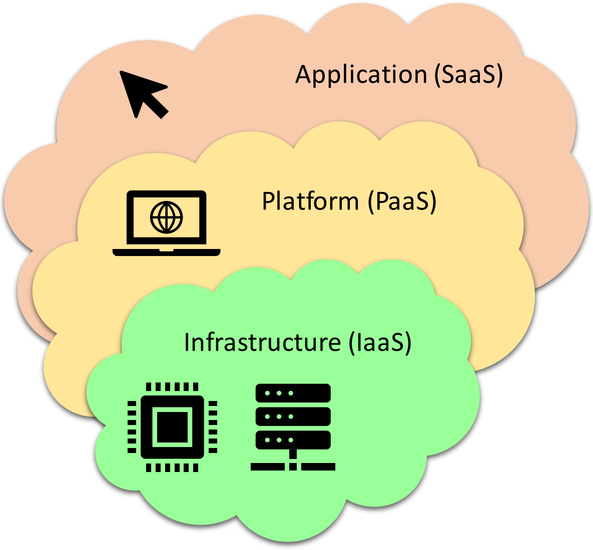


Figure 1: Graphical representation of IaaS, PaaS, and SaaS models

The SaaS model has also become increasingly popular in the recent years both in enterprise and individual user space, where hosting and accessing cloud-based software is preferred over having to install and maintain a specific software suite on local machines, dealing with constant updates, security patches, and licensing. In the SaaS model, the client effectively leases the specific application(s) over extended period of time, making use of it over a public Internet or private Intranet connection, and optionally releasing the associated licenses once the work has been completed. This model allows enterprises to save on licensing costs for more expensive application packages, especially with the so-called floating licensing models, where licenses are checked out from a centralized licensing server on demand and returned to the pool once the application is closed. For individual clients, the SaaS models provide ability to access the given application using a variety of platforms (operating systems) using nothing more than their preferred Internet browser, and get a consistent behavior anywhere and at any time, without having to worry about installing and maintaining the given application up to date across multiple hardware and software platforms.

2.3 Risks and vulnerabilities

Apart from obvious advantages, there are also risks [5], [8], [5], [8] associated with cloud computing, namely cloud deployment models [1],and all the delivery models described before.

Security and privacy [5] are the two biggest concerns since data and infrastructure management in the cloud is provided by a third-party, with potential interest in obtaining illicit access to specific content for nefarious reasons. Therefore, there is always a security risk associated with sharing sensitive information with the cloud service providers. Although the cloud computing vendors ensure highly secured password protected accounts, access to the given cloud application may be breached, resulting in illicit access to the given application and resources. Separating roles and responsibilities and maintaining such sensitive information within the given enterprise-owned and maintained IT infrastructure may address such concerns, at least partially.

The failure of the isolation mechanism separating storage, memory, and routing between the different tenants have become infamous these days, primarily due to the variety of isolation attacks at the hardware level (CPU, memory, motherboard, etc.). Complex hardware-level attacks (e.g., Meltdown [9] - CVE-2017-5754, Spectre [10] - CVE-2017-5753 and CVE-2017-5715, affecting Intel and AMD CPUs) allow malicious actors access content from other virtual machines sharing the same hardware platforms managed by cloud computing providers. New security flaws are exposed with some regularity, leading to increasing concerns about storing sensitive information with cloud-based providers, or any cloud-based platforms – even if they are owned and managed by individual enterprises.

There is also the risk associated with the ability to recover data after insecure or incomplete data deletion [5], [11]. It is not likely a big concern for individual users in SaaS scenarios, where content is maintained in a virtualized environment and not accessible at the hardware level. It is, though, a concern for enterprise customers using IaaS model, where data stored on leased physical storage infrastructure may become accessible to other users once the storage is recycled for other purposes.

It is clear, therefore, that cloud computing has a lot of advantages and disadvantages, especially for enterprise customers requiring stricter security control for their content and resources. For individual users focused on SaaS like applications, though, cloud computing provides plenty of advantages, primarily associated with application mobility across different operating systems, hardware and software platforms, persistent access from anywhere in the world using just Internet access, and inherent security associated with not having to maintain the given application updated and patches for security reasons. The cost of these advantages is obviously entrusting the given content to the cloud provider, something that most individual SaaS model users do not seem to mind.

2.4 Related Work on SaaS

"*Cloud computing brings new issues, challenges, and needs in performance testing, evaluation and scalability measurement due to the special features of cloud computing, such as elasticity and scalability*" [12]. The majority of studies to date addressing these challenges of the SaaS model [12], [13], [14], [15], [16] have been primarily focused on the SaaS host side performance and scalability.

Another study area for SaaS is related with the architecture, where SaaS customization strategies (including at the PaaS level) are addressed, and where the multi-tenancy architecture affects security isolation, performance, and remains an engineering challenge [17], [18], [19], [20]. SaaS scalability mechanisms include a multi-level architecture with load balancers, automated data migration, and software design strategies [21].

Other studies focus on the SaaS adoption from the perspective of enterprise (organization) [22] and individual users, acknowledging the recent growth observed for SaaS and associated products. It is noted, though, that some organizational users are open to SaaS and its use in enterprise environment, while others are still hesitant. Security-related concerns are quoted as the most prevalent reason for delaying SaaS adoption, preferring to maintain the use of the standard local IT-based support models instead. It is also noted that security breaches in the SaaS infrastructure and associated cost might make some of the enterprise SaaS adopters reconsider their strategies and move back to a local IT-based model.

Google Docs is used as one of the more popular examples of SaaS platforms today for editing documents, both by enterprise and individual users. Therefore, studies focusing on identifying and understanding factors that influence users' acceptance of SaaS collaboration tools in organizational setting [23] are becoming a requirement not only to understand the process of decision-making but also to understand the trends and constraints.

The impact of packet loss and individual latency components associated with Google Docs has also been examined [24], with an interesting linear regression model proposed to derive the application performance for the given network type, latency, and packet loss ratios.

Finally, security related aspects associated with SaaS have received a lot of attention, focusing on cloud computing security risks [7], potential attack vectors [8], and user requirements, while also attempting to create a viable solution that eliminates these potential threats [25], [26], thereby improving the SaaS experience for security-conscious enterprise customers. The proposed solutions typically rely on cryptography, specifically Public Key Infrastructure, or ensure that data stored in the cloud environment is safe and can be accessed only by authorized agents.

2.5 Open issues

Even though there have been some studies proposing stratified cloud monitoring [27] with proposed metrics across several layers [28] and case studies for storage services [29][30][31], to the best of our knowledge, SaaS has not been examined to date in terms of the following aspects:

- SaaS behavior across different operating system, validating claims that SaaS applications perform equally well irrespective of the operating systems in use;
- SaaS behavior across different Internet browsers, validating claims that SaaS applications perform equally well irrespective of the browser in use;
- SaaS behavior is geographically independent, i.e., its behavior is the same irrespective of the location it is being accessed from;
- SaaS behavior is independent from the access network performance and type, with the special focus on user mobility and the use of mobile networks;
- IPv6 readiness of SaaS solutions available to individual users, critical for the long-term adoption of not only SaaS but also IPv6 in individual and enterprise networks; and
- QoE-specific analysis, by using QoS metrics, for specific SaaS specific applications and in the context of cloud computing [32][33][34].

The aforementioned aspects of the SaaS operation and interaction with the end-user are examined in the following section, where a number of hypotheses related with the SaaS and its operation are presented and examined in detail.

3

WORK AND SCOPE

This chapter describes the work and scope of this thesis, which aims at analyzing and observing the SaaS behavior as well as the impact of SaaS operation on the local network, in different scenarios, network configurations, operating systems, and browsers. Data collected in individual scenarios (packet traces and screen recordings) are then used to assess the validity of different hypotheses puts forth before the start of this study. Individual collected packet traces were processed to exclude any SaaS unrelated traffic (LAN local traffic, including for example ARP, link local traffic, multicast, IPv6 traffic in cases where no IPv6 communication with SaaS hosts was expected, etc.), and averaged to minimize the impact of any scenario outliers.

3.1 Hypotheses

In order to analyze the impact that SaaS applications have on the network, and as a starting point on the assessment of the network performance, some hypotheses are formulated.

The hypotheses are described in this section and are examined in detail in Section 4 of this thesis, through the analysis of specific SaaS for particular combinations of operating systems (Windows 7, Windows 10, Kubuntu), browsers (Google Chrome, Mozilla Firefox, and Internet Explorer – where supported), and different network configurations (see Section 3.2 for details). Individual hypotheses may be proved or disproved in the course of the scenario analysis, with details presented in the respective sections.

3.1.1 Hypothesis A

Events on SaaS applications, namely major events (inserting large blocks of text, images, tables, animations, functions or charts), can be correlated to specific traffic patterns, i.e., activity peaks.

3.1.2 Hypothesis B

The SaaS application is delivered to end user over TCP.

3.1.3 Hypothesis C

The SaaS application generates more upstream than downstream traffic.

3.1.4 Hypothesis D

The SaaS application has consistent behavior for the same browser across all examined operating systems, i.e., it is operating system independent.

3.1.5 Hypothesis E

The SaaS application has consistent behavior for an operating system for all examined browsers, i.e., it is browser independent.

3.1.6 Hypothesis F

The SaaS application has consistent behavior for an operating system and for a browser in different network scenarios, i.e., it is network configuration independent.

3.1.7 Hypothesis G

The SaaS application has preference for IPv6 communication when available, i.e., when both IPv4 and IPv6 routing is available, only IPv6 traffic is exchanged with the SaaS host.

3.1.8 Hypothesis H

The Quality of Experience (QoE) for the same SaaS application is network-independent, i.e., the end user experience for the same application remains the same, irrespective of the network type used to access the given SaaS application.

3.2 Network Configurations

Individual hypotheses have been examined in three different network configurations described in the following sections.

3.2.1 Network Configuration 1 (NC1)

The NC1 scenario features a home local area network (LAN) with Fiber to the Curb (FTTC) and DOCSIS 3.0 (Data Over Cable Service Interface Specification) based access to the Internet. The service from the

Internet Service Provider (ISP) features 120 Mbps downstream and 10 Mbps upstream (both maximum values), as defined in their Service Level Agreement (SLA). The highly asymmetric data rate of the offered service is expected to have some impact on the examined applications, especially in terms of the upload speeds.

The LAN configuration is shown in Figure 2, comprising the following elements: the ISP provided cable modem with embedded wireless access, connected to a second-level aggregation (access) gigabit Ethernet switch, a SIP-based IP phone, a network printer, a desktop class PC, a network-connected smart TV, and a laptop used for target tests, connecting to the rest of the network infrastructure via WiFi.

A dual-stage network switching is used: the ISP provided cable modem with embedded wireless gateway offers first stage switching and wireless connectivity; and a wired 1 Gbps Ethernet switch, which offers second stage switching, providing additional port capacity needed in this LAN design.

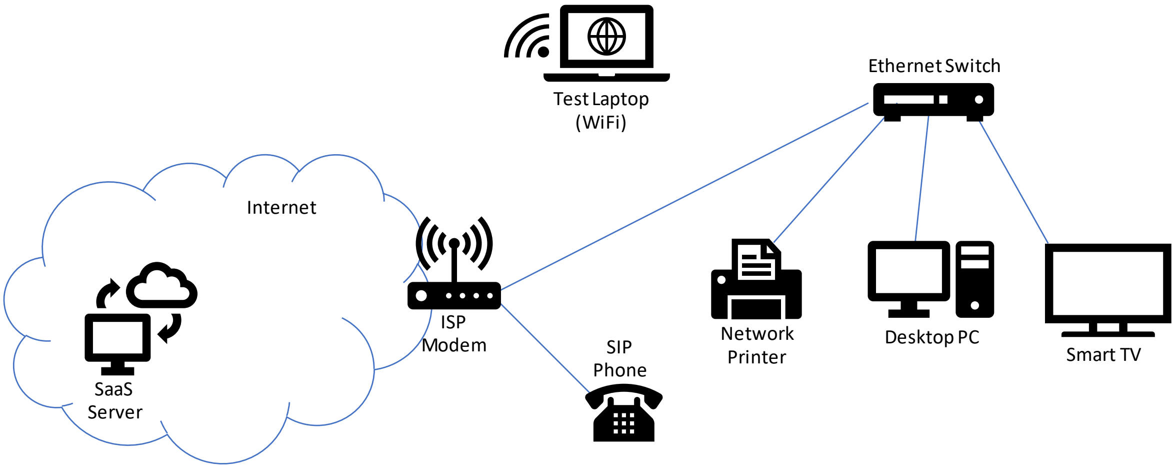


Figure 2: Network Scenario 1

3.2.2 Network Configuration 2 (NC2)

The NC2 scenario features a LAN with fiber-based access to the Internet (FTTH – Fiber to the Home). The service from the ISP, as specified in their SLA, features a symmetric 1 Gbps service (1000 Mbps downstream and upstream rates). Given the symmetric data rate in downstream and upstream directions, no bottleneck in the LAN is expected in this scenario.

The LAN configuration is shown in Figure 2, comprising the following elements: the core gigabit Ethernet switch, connected to several second-level aggregation gigabit Ethernet switches. Each access switch has a dual antennae WiFi Access Point (AP) connected to it, providing distributed WiFi for best coverage. All APs are connected to the core Ethernet switch via 1 Gbps Ethernet links and have been validated to support the sustained data rate in excess of 800 Mbps. The LAN comprises also several smart TVs, personal computers, smart devices (Internet of Things – IoT), a centralized data storage server hosting a number of virtual machines, and others. The test laptop connects to one WiFi AP during the whole test session and does not roam between multiple APs to provide optimum network connectivity and throughput.

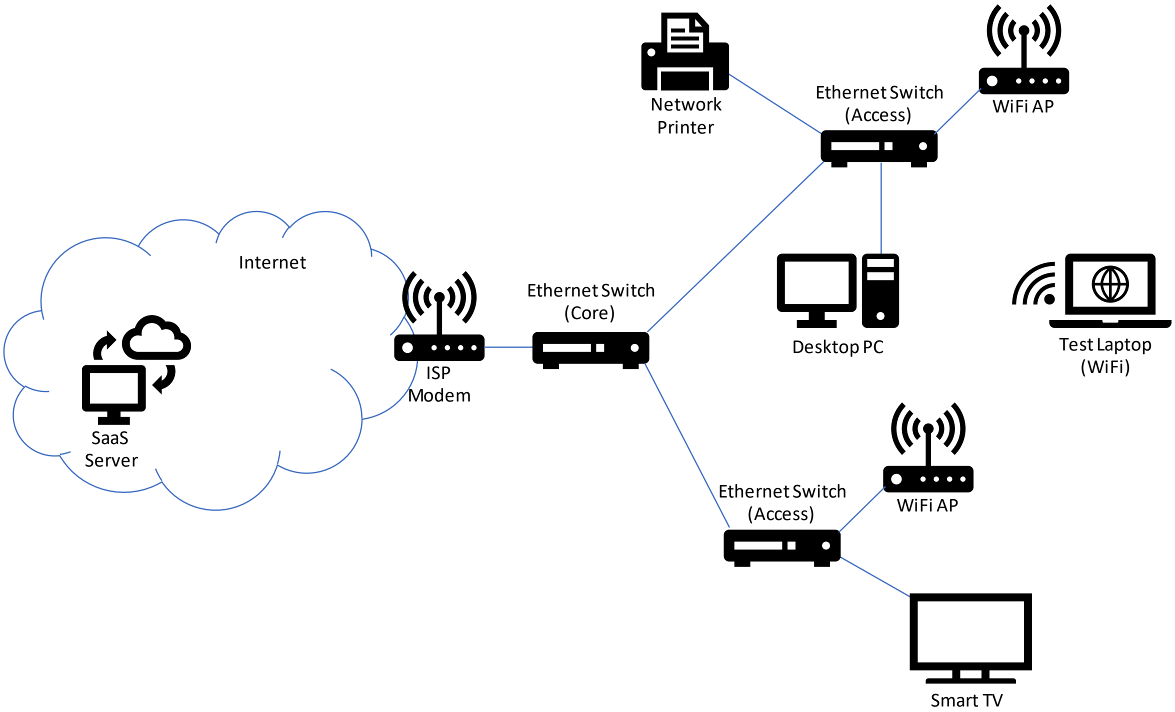


Figure 3: Network Scenario 2

3.2.3 Network Configuration 3 (NC3)

The NC3 scenario features a personal area network (PAN) created by a fully roaming test laptop connected to a T-Mobile LTE network for data connectivity. The following bands are used by the LTE modem: Band 2 (1900 MHz), Band 4 (1700/2100 MHz), Band 5 (850 MHz), Band 12 (700 MHz), Band 66 (Extension of band 4 on 1700/2100 MHz), Band 71 (600 MHz). The resulting PAN is shown in Figure 3. The data rate and latency values, as defined in the SLA, are as follows (minimum guaranteed – maximum achievable):

- Download speeds: between 9 – 47 Mbps (typical values)
- Upload speeds: between 4 – 20 Mbps (typical values)
- Latency: between 30 – 50 ms (typical values)

There are no additional devices in the PAN. The LTE Hotspot was placed in a location providing optimum LTE signal coverage, optimizing both download and upload speeds.

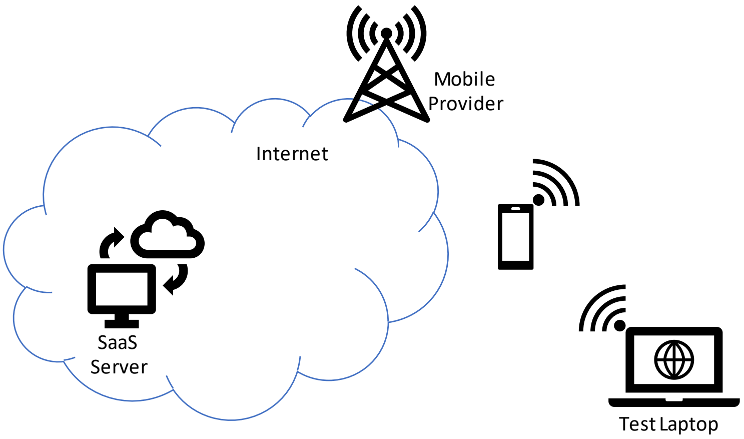


Figure 4: Network Scenario 3

3.2.4 Network Addressing

NC1 and NC2 use one of the default private IPv4 (Internet Protocol, version 4) address subnets of 192.168.1.0/24 [35], with a few IPv4 addresses pre-assigned to specific devices/functions, for example: the network gateway is at 192.168.1.1, the printer is at 192.168.1.199 and the server is at 192.168.1.200. A block of addresses is reserved for DHCP-based assignment, while all well-known and trusted devices are assigned fixed (pre-defined) addresses via DHCP, pulling from a reserved pool of addresses. For security reasons, all IoT devices, if present, are isolated in a separate Virtual LAN (VLAN) with a dedicated IPv4 subnet. No IPv6 is used in NC1 and NC2 scenarios, allowing to eliminate any observed IPv6 traffic as background.

NC3 relies on the LTE network, and by default the mobile carrier (T-Mobile) allocates both IPv4 and IPv6 public addresses when using the respective network for Internet connectivity. Based on initial test traces collected, routing preference is given to IPv6, though limited IPv4 communication was also observed in some cases. This means that for NC3, IPv6 cannot be excluded and more specific filters (likely, focusing on target address scope for the SaaS provider) need to be designed.

3.3 Platforms and Tools

The following platforms and operating systems were used in this study:

- Operating Systems:
  - Microsoft Windows 7 Pro, 64-bit, used in NC1 and NC2 scenarios;
  - Microsoft Windows 10 Pro, 64-bit, used in NC2 and NC3 scenarios;
  - Kubuntu Linux 19.10, 64- bit, used in NC2 and NC3 scenarios.

- Browsers: Google Chrome (GC); Mozilla Firefox (FF); and Internet Explorer (IE). The latest versions of individual browsers available at the time when the study was started were used, with the underlying operating systems up to date.
- SaaS platform: Google Docs (GD) and Microsoft Office Online (WO)
- Packet capture and analysis: Wireshark 2.0.4 with TShark extensions
- Screen recording software:
  - TechSmith Snagit Editor, version 12.1.0, under Windows OS;
  - SimpleScreenRecorder, version 0.3.11, under Kubuntu OS.
- Macro actions recording software:
  - Mouse and Keyboard Recorder, version 3.2.8.6;
  - Xnee, version 3.19-3.

A conceptual test scenario is shown in Figure 5, where a test laptop is connected using one of the available access network solutions to at least one SaaS host via public Internet. The test laptop is running the end-user SaaS application and it is also used to perform packet captures, screen recordings, etc., representing the data collection point. The said data collection can be done via direct user interaction with all the applications (path represented with the blue unidirectional arrows) or by using actions recorder software (path represented with the red unidirectional arrows), which in turn will interact with the applications the user would in normal circumstances.

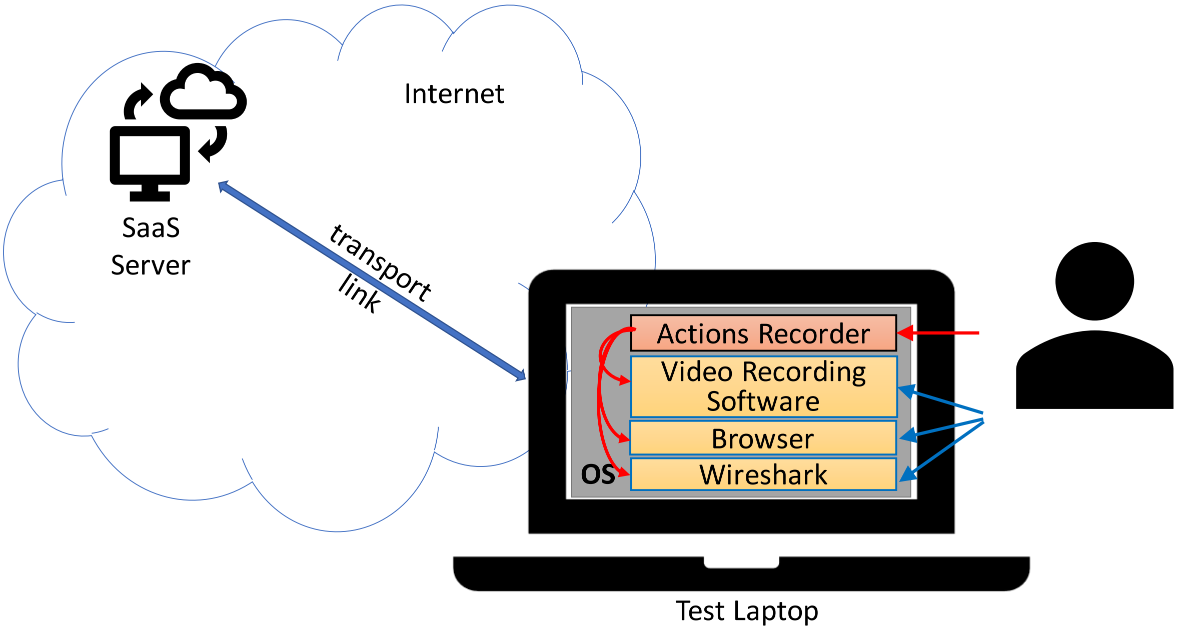


Figure 5: Conceptual test scenario

3.4 Scenario Naming

To keep the description of individual scenarios concise and unambiguous, a specific naming strategy was employed, marking the specific SaaS type under consideration, operating system, network configuration, and browser. The following naming sequence is therefore used:

<NetworkConfiguration><OperatingSystem><SaaS><Browser>

- <NetworkConfiguration>: indicates one of the predefined network configurations: NC1, NC2, or NC3, as described in section 3.2;
- <OperatingSystem>: indicates the operating system used for the analysis, namely: Windows 7 (W7), Windows 10 (W10), or Kubuntu (KU);
- <SaaS>: indicates the type of SaaS used, namely: Google Docs (GD), WO;
- <Browser>: indicates the browser used for the analysis, namely Google Chrome (GC), Mozilla Firefox (FF), or Internet Explorer (IE).

For example, "NC2W7GDFF" implies scenario tested in network configuration 2, using W7, Google Docs SaaS and FF. The same naming scheme can be used to designate a more generic test scheme, for example all scenarios tested in network configuration 2, using W10, irrespective of the browser or the SaaS itself, would be designated as NC2W10.

3.5 Tests Taxonomy

In order to provide a clear and comprehensive overview of the tests executed during the research work of this thesis, as well as its scope extent, a taxonomy of tests is summarized on Table 1. The table lists the executed tests, their features comprising the variability of software in which the execution took place and as well a brief description of the reason for their execution.

Table 1: Tests taxonomy

Test Scenario – Name	Features							Goal
	OS		**Browser**			**SaaS**		
	W	*Ku*	*FF*	*GC*	*IE*	*Google Suite*	*Office Suite*	
NC1W7GDFF	7		X			GD		1) Compare and assess the general behavior of each SaaS under different browsers 2) Check correlation between events and
NC1W7GDGC	7			X		GD		
NC1W7GDIE	7				X	GD		
NC1W7WOFF	7		X				WO	
NC1W7WOGC	7			X			WO	
NC1W7WOIE	7				X		WO	
NC1W7GSFF	7		X			GS		

NC1W7GSGC	7			X		GS		traffic patterns during execution of tasks
NC1W7GSIE	7				X	GS		
NC1W7POFF	7		X				PO	
NC1W7POGC	7			X			PO	
NC1W7POIE	7				X		PO	
NC1W7GXFF	7		X			GX		
NC1W7GXGC	7			X		GX		
NC1W7GXIE	7				X	GX		
NC1W7XOFF	7		X				XO	
NC1W7XOGC	7			X			XO	
NC1W7XOIE	7				X		XO	
NC1W7GHFF	7		X			GH		
NC1W7GHGC	7			X		GH		
NC1W7GHIE	7				X	GH		
NC1W7SOFF	7		X				SO	
NC1W7SOGC	7			X			SO	
NC1W7SOIE	7				X		SO	
NC2W7GDFF	7		X			GD		1) Further analysis of SaaS usage impact on network and validation of behavior consistency 2) Compare and assess the general behavior for one specific application – text editing, from each SaaS under different browsers and OS for different networks configurations (NC2 and NC3) 3) Verify the suitability of the different SaaS text editing application, for a mobile LTE connection (NC3 only)
NC2W7GDGC	7			X		GD		
NC2W7GDIE	7				X	GD		
NC2W7WOFF	7		X				WO	
NC2W7WOGC	7			X			WO	
NC2W7WOIE	7				X		WO	
NC2W10GDFF	10		X			GD		
NC2W10GDGC	10			X		GD		
NC2W10GDIE	10				X	GD		
NC2W10WOFF	10		X				WO	
NC2W10WOGC	10			X			WO	
NC2W10WOIE	10				X		WO	
NC2KUGDFF		X	X			GD		
NC2KUGDGC		X		X		GD		
NC2KUWOFF		X	X				WO	
NC2KUWOGC		X		X			WO	
NC3W7GDFF	7		X			GD		
NC3W7GDGC	7			X		GD		
NC3W7GDIE	7				X	GD		
NC3W7WOFF	7		X				WO	
NC3W7WOGC	7			X			WO	
NC3W7WOIE	7				X		WO	
NC3W10GDFF	10		X			GD		
NC3W10GDGC	10			X		GD		
NC3W10GDIE	10				X	GD		
NC3W10WOFF	10		X				WO	
NC3W10WOGC	10			X			WO	
NC3W10WOIE	10				X		WO	
NC3KUGDFF		X	X			GD		
NC3KUGDGC		X		X		GD		
NC3KUWOFF		X	X				WO	
NC3KUWOGC		X		X			WO	

3.6 Trace Analysis Methodology

In each examined combination of network configuration, OS, browser, and SaaS, several (at least 7) repetitions of each scenario were carried out, collecting the packet trace on the test laptop as well as the screen capture of the resulting scenario execution. Captures done under NC1 are the only exception, with a single packet capture performed.

The collected traces were then averaged, discarding any IP address information and sorting in decreasing order of packet / byte count per SaaS host address. In this way, traces are normalized [36] in the function of the SaaS host addresses in the decreasing order of importance, calculating the average packet and byte counts per SaaS host address for the aggregate (combined upstream and downstream) as well as separated upstream and downstream directions, needed for further analysis. The methodology used for each and every network configuration and scenario are shown below (see Table 2) using the NC2W7GDFF scenario as an example, with 8 collected trace samples.

First, each collected trace is filtered to discard any local LAN/PAN traffic, by applying respective filters described in Appendix A. These filters are intended to discard all local SaaS-unrelated traffic (ARP, DNS, IPv6 traffic, etc.) as well as any public IPv4 traffic that is unrelated with given SaaS under consideration. For example, when examining GD scenarios, it is obvious that only public IP addresses belonging to Google are of interest, discarding any communication with (for example) Amazon, Microsoft, Akami, etc. Such background communication, while typically low in volume, might influence observations, providing potentially false conclusions.

Table 2: NC2W7GDFF scenario, aggregate packet count, source data

Average	Samples								Host
	#1	#2	#3	#4	#5	#6	#7	#8	#
3988	4107	3984	4107	3923	3935	3934	4060	3853	**1**
1205	1184	1452	1184	1105	1168	1094	1225	1228	**2**
546	1004	624	1004	237	221	376	688	217	**3**
199	140	604	140	168	115	170	126	125	**4**
102	102	165	102	84	52	145	111	55	**5**
67	48	136	48	67	43	95	63	39	**6**
50	44	96	44	44	39	50	46	35	**7**
43	40	69	40	38	36	44	46	31	**8**
34	31	47	31	35	32	32	34	30	**9**
32	30	42	30	33	27	31	32	29	**10**
27	29	36	29	31	27	3	32	28	**11**
19	7	36	7	29	3		27	26	**12**
11		7		23			7	7	**13**
3		3					3		**14**

Once the trace has been filtered for SaaS-unrelated local and public IPv4 traffic, aggregate, upstream, and downstream statistics for each trace are collected and averaged. The Table 2, for example, shows the resulting data for aggregate packet count using the NC2W7GDFF scenario. As indicated before, 8 sample traces were examined, with the resulting average values shown in the first column and highlighted in green. Note that the number of individual SaaS hosts in each collected packet capture may be different, depending on the specific SaaS, time of the day network load, etc.

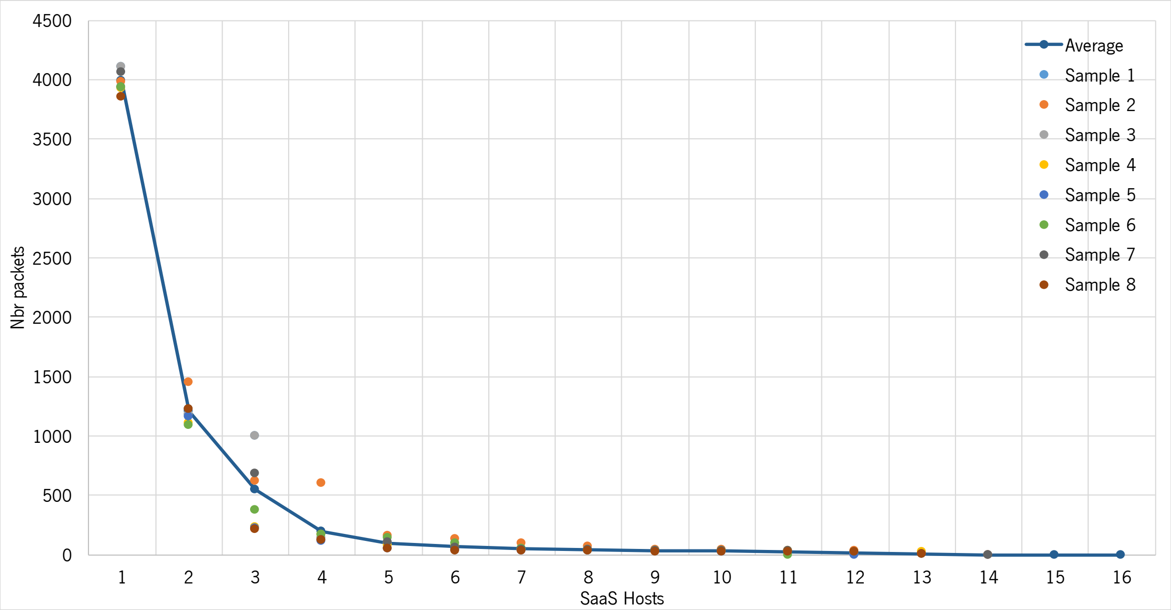


Figure 6: NC2W7GDFF scenario, aggregate packet count, source and average data

Figure 6 shows the plot of individual data samples as well as the fitting of the average packet count in the function of the SaaS host address.

Similar curves and the associated data tables are collected for packet and byte count for the aggregate, upstream, and downstream directions, providing a complete averaged data trace for the given scenario under consideration.

The process of averaging the collected packet traces eliminated any outliers and allows for higher confidence analysis. Therefore, the comparison between individual traces are free of any potential impact of outlying variations from the collected set of packet traces.

4

HYPOTHESES AND THEIR ANALYSIS

This chapter focuses on the analysis of individual hypotheses outlined in Section 3.1. Additional details are provided for the reasoning behind each hypothesis, expectations towards each scenario, as well as the comparison of these expectations and the observations based on the set of individual examined scenarios. Where possible, all operating systems, network configurations, browsers, and SaaS combinations are examined and the summary is presented in a simple two-dimensional table, simpler to analyze and summarize. Examples of results proving and disproving (where applicable) the given hypothesis are presented, focusing on more interesting cases, and avoiding repetition of results that are similar, primarily to avoid a very extensive document and to improve readability.

4.1 Hypothesis A: Positive correlation between SaaS application events and traffic patterns

Various types of events (inserting large blocks of text, images, tables, animations, functions or charts) when using the given SaaS platform are expected to generate a certain level of network activity (peaks) that can be correlated when comparing the packet capture and the screen recording for the given session. Effectively, such a positive correlation between the SaaS events and the network activity allow to predict the behavior of specific SaaS applications under certain conditions, i.e., on congested LAN, in mobility scenarios with LTE uplink to the Internet, etc.

Due to the software availability and network access at the time this hypothesis was being examined, only a limited set of tests was concluded. The tests were performed covering only W7 Pro 64-bit operating system, the only considered at the time, but with the intent of examining the behavior of several SaaS applications (GD, GS and GX for Google Docs, and WO, PO and XO for Microsoft Office Online). All the three browsers – FF, GC and IE, were examined in this scenario to confirm whether there are any observable differences between them.

4.1.1 NC1W7GD and NC1W7WO scenarios

This section examines the operation of SaaS Google Docs (GD) for reference browsers, i.e., GC, FF, and IE: NC1W7GDFF; NC1W7GDGC and NC1W7GDIE, based on Wireshark captures.

In these scenarios, lasting between 4 to 5 minutes, a user creates and edits a text document with the resulting size of approximately 120 KB (for reference, when saved locally using Microsoft Office Word in docx format), executing the following basic tasks:

- creating and naming a new document;
- inserting, editing, and formatting specific text; and
- inserting and manipulating images and tables.

The filters applied to the collected packet captures are presented in Appendix A, following the process described in Section 3.6. The final number of packets, after applying the respective filters, for each scenario is as follows: NC1W7GDFF – 3774 packets; NC1W7GDGC – 4290 packets, and NC1W7GDIE – 5263 packets. It is immediately visible that the NC1W7GDIE scenario – even though the shortest of all examined scenarios – is also most packet-rich, at about 23% packets more when compared to the NC1W7GDGC scenario and at about 39.5% packet more when compared to the NC1W7GDFF scenario. The packet count in scenarios NC1W7GDFF and NC1W7GDGC is roughly similar.

The general behavior of data exchanged (Bytes) under NC1W7GDFF is presented in Figure 7 (top, using linear scale) and Figure 7 (bottom, using logarithmic scale).

Analyzing the charts for NC1W7GDFF, most of the time there are bursts of data being exchanged, that are smaller or close to 1 KB. The volume of data received (as shown in blue, Figure 7) is slightly higher, except for moments of high upload/modification of information. Considering that SaaS Google Docs implementation is not open source it is only possible to speculate as to why this happens. This behavior could possibly be explained with the implementation of the SaaS platform that has to constantly update the information to the end user on the browser and to the fact that the software, in order to provide reliability to the end user, needs to save the information and at the same time reflect the changes done on the document. Similar behavior was verified for NC1W7GDGC, being drawn the identical conclusions.

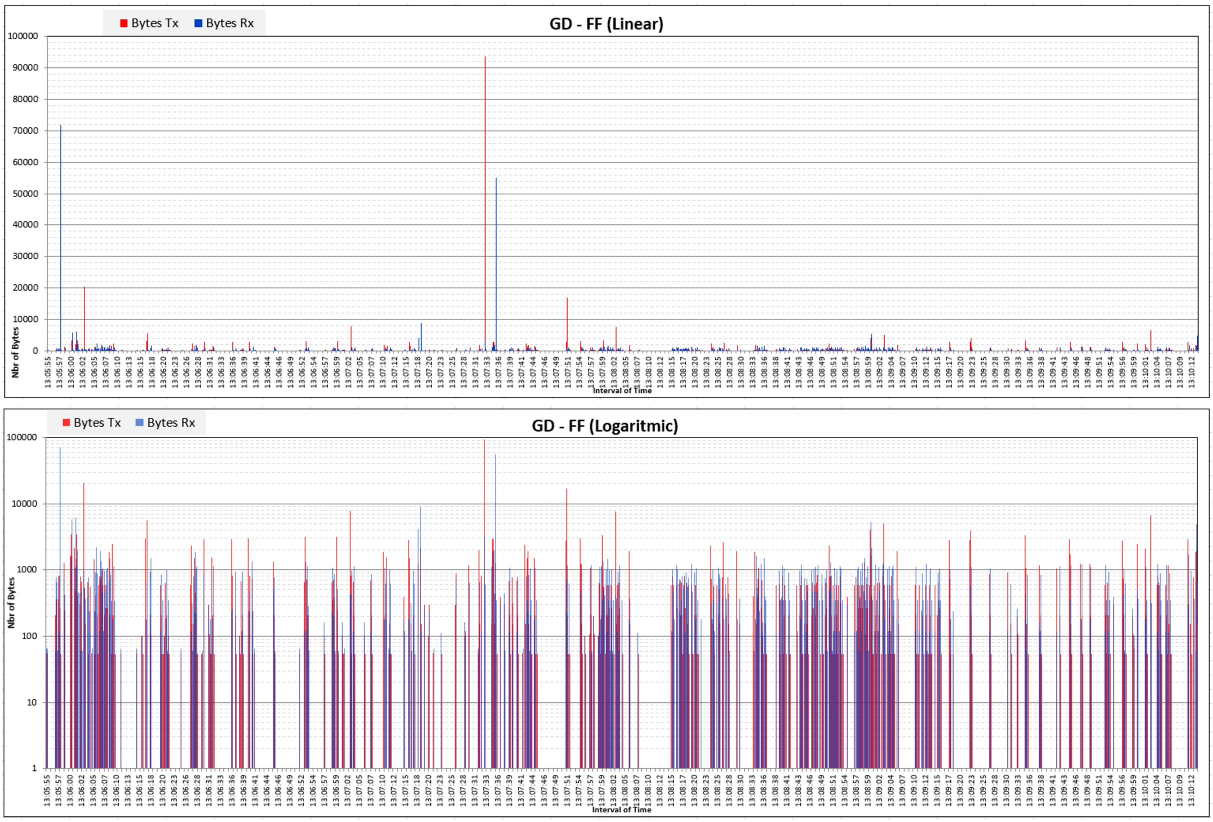


Figure 7: NC1W7GDFF - bytes exchanged in the filtered packet trace, using linear scale (top) and logarithmic scale (bottom)

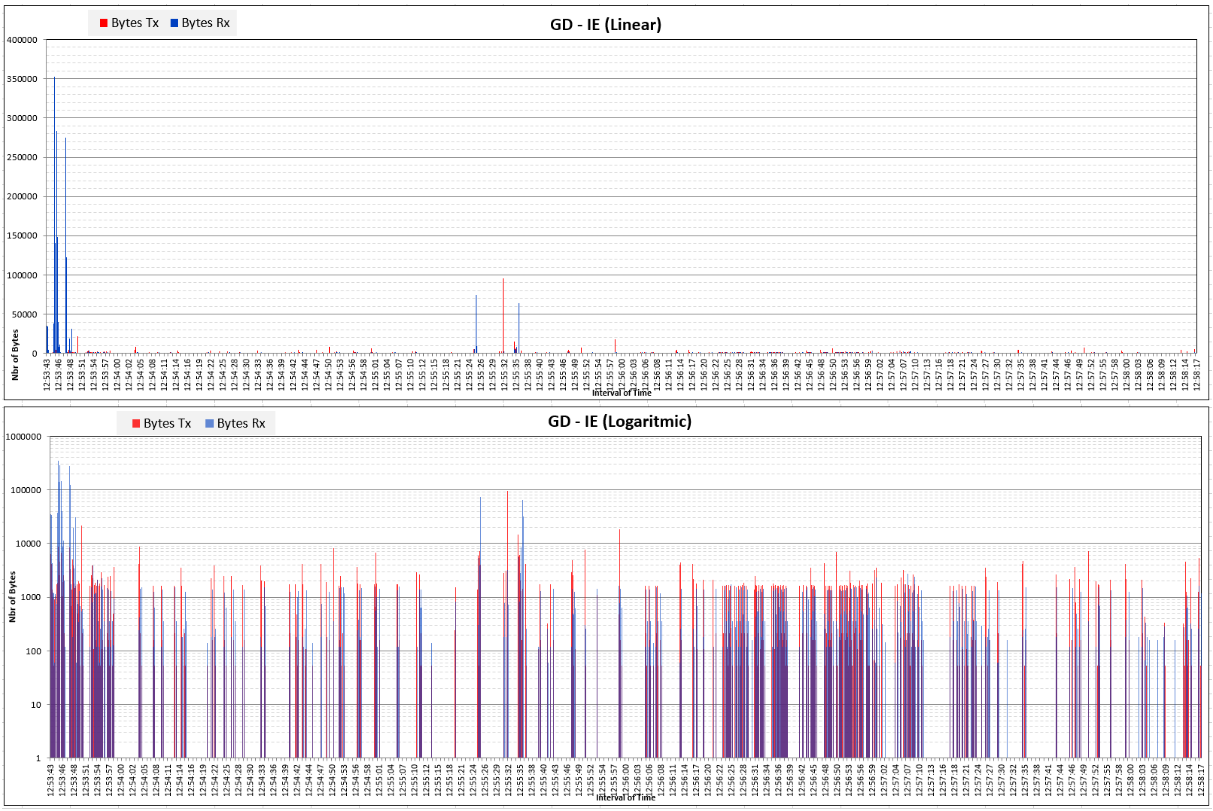


Figure 8: NC1W7GDIE - bytes exchanged in the filtered packet trace, using linear scale (top) and logarithmic scale (bottom)

The observed general behavior in the NC1W7GDIE scenarios is shown in Figure 8 and it is noticeably different from both NC1W7GDFF and NC1W7GDGC scenarios, in what concerns the initial download-intensive operation phase (likely, the initial SaaS GUI download). After that the volume of the downstream and upstream traffic is comparable even when examined at the logarithmic scale. This initial behavior stands in contrast with NC1W7GDFF and NC1W7GDGC scenarios, where the SaaS GUI does not seem to have such a large impact on the overall packet trace size.

This conclusion is further substantiated by examining shown in Table 3 for the NC1W7GDIE, where the ratio between downstream (Rx) and upstream (Tx) traffic is equal to 3 (versus approximately 1 for NC1W7GDFF and NC1W7GDGC scenarios). The similar ratio is also observed in terms of the average data rate (downstream versus upstream), while the peak rate ratio is higher (close to 6.25), indicating high received rate when compared with the transmitted data rate. Summary traffic profiles for NC1W7GDFF and NC1W7GDGC scenarios are shown in Table 4 and Table 5, respectively, exhibiting substantial similarities for both FF and GC browsers for SaaS GD.

Table 3: NC1W7GDIE summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	2153.46	KB
Total Data Tx	716.65	KB
Rx rate, avg	64.15	kbps
Tx rate, avg	21.35	kbps
Total (Tx + Rx) rate, avg	0.085	Mbps
Rx rate, max	5083.78	kbps
Tx rate, max	814.76	kbps

Data Volume Ratio (Rx/Tx)
3.00

Ratio of Avg Data rate (Rx/Tx)
3.00

Ratio of Max Data rate (Rx/Tx)
6.24

Table 4: NC1W7GDFF summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	469.81	KB
Total Data Tx	438.62	KB
Rx rate, avg	14.86	kbps
Tx rate, avg	13.87	kbps
Total (Tx + Rx) rate, avg	0.029	Mbps
Rx rate, max	594.55	kbps
Tx rate, max	748.98	kbps

Data Volume Ratio (Rx/Tx)
1.07

Ratio of Avg Data rate (Rx/Tx)
1.07

Ratio of Max Data rate (Rx/Tx)
0.79

Table 5: NC1W7GDGC summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	543.68	KB
Total Data Tx	492.28	KB
Rx rate, avg	17.33	kbps
Tx rate, avg	15.69	kbps
Total (Tx + Rx) rate, avg	0.033	Mbps
Rx rate, max	778.98	kbps
Tx rate, max	761.52	kbps

Data Volume Ratio (Rx/Tx)
1.10

Ratio of Avg Data rate (Rx/Tx)
1.10

Ratio of Max Data rate (Rx/Tx)
1.02

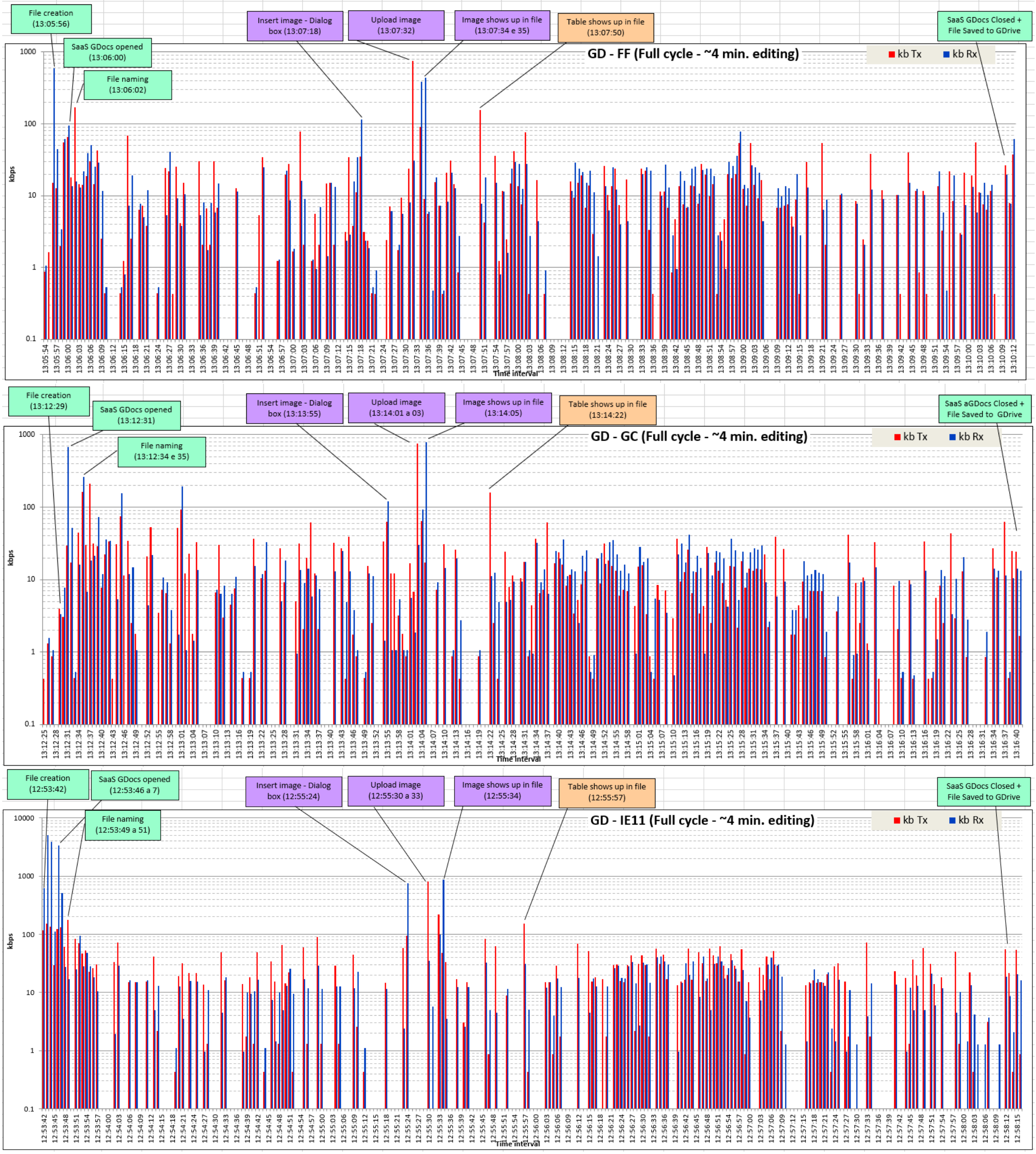


Figure 9: Correlation between throughput and chronological order of major events (in logarithmic scale) for NC1W7GDFF, NC1W7GDGC, and NC1W7GDIE (from top to bottom)

Through the comparison of the recorded screen capture session and the packet trace for each and every test session, it was possible to correlate individual data bursts and several events related with the creation and editing of the text document. The event correlation is shown in Figure 9 for NC1W7GDFF, NC1W7GDGC, and NC1W7GDIE scenarios (top to bottom).

Looking at the event and network activity correlation shown in Figure 9**,** it is clear that the processes of file creation, naming, and loading the GD GUI in the browser represent one of the biggest data transfer only comparable to the process of inserting images or tables in the file, for all examined browsers. The insertion of images is, though, the most data-heavy process since it implies not only an upload of information to the SaaS host(s) but as well as an accurate display of the document being edited on the end user monitor, therefore being a major data transfer both in upstream and in downstream directions. It can be concluded, though, that the NC1W7GDIE scenario features the most network-intensive operation at the beginning of the scenario, where the GD GUI is being loaded and displayed on the end user client.

For completeness, the correlation between minor (simpler) actions and the resulting network activity is shown in Figure 10, but only for NC1W7GDFF (being NC1W7GDGC almost identical to NC1W7GDFF) and NC1W7GDIE (top to bottom).

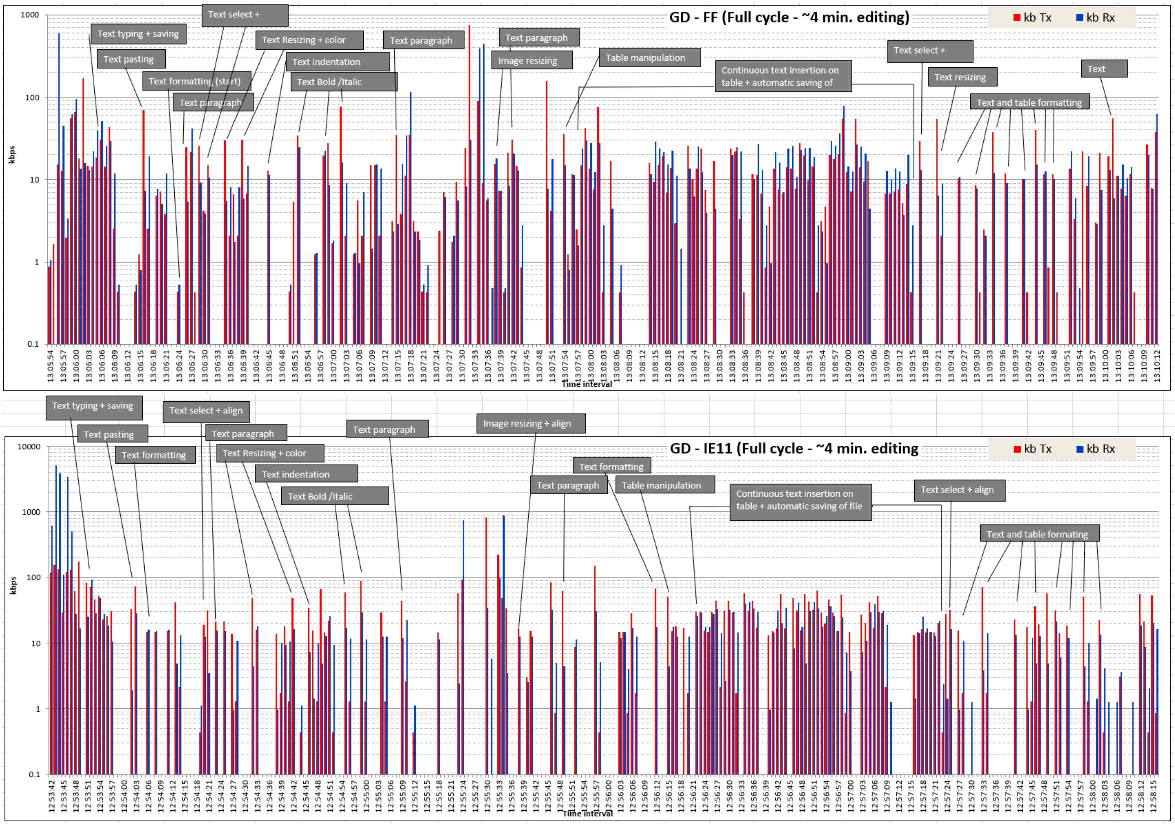


Figure 10: Correlation between throughput and chronological order of minor events (in logarithmic scale) for NC1W7GDFF and NC1W7GDIE (from top to bottom)

In general, considering the event and network activity correlation shown in Figure 9 and in Figure 10, we can observe that the traffic profile is very similar for NC1W7GDFF, NC1W7GDGC, and NC1W7GDIE, being the 2 first ones almost identical (in terms of visual traffic profile). The moments of continuous text insertion and formatting coincide with periodic network activity displaying small bursts of data at a data rate below 100 kbps. Even though these tasks do not cause heavy network activity, which is mainly visible when loading the GUI and inserting objects (images, tables, etc.), they still need to be taken into the account in the network bandwidth management due to its persistent character for this application.

This same analysis methodology used for SaaS GD was then applied to SaaS Microsoft Word Online (WO) using the same reference browsers, providing the following scenarios: NC1W7WOFF, NC1W7WOGC, and NC1W7WOIE, based on captures with approximate duration between 5 to 5.30 minutes.

The final number of packets for each filtered scenario are as follows: NC1W7WOFF – 7013 packets, NC1W7WOGC – 6301 packets, and NC1W7WOIE – 6270 packets. Although there is a general increase in terms of number of packets (which can be due to the trace duration and to the way the SaaS WO operates), all the examined browsers perform similarly in this scenario. Curiously, in this particular test scenario, IE is least network intensive, with the smallest number of packets from the all three scenarios.

The general behavior of data exchanged for all the three browsers is similar and, therefore, the following descriptions and conclusions stand for all of them, and NC1W7WOIE will be used as an example.

The example of NC1W7WOIE, shown in Figure 11 (top, using linear scale) and Figure 11 (bottom, using logarithmic scale) is used to show the traffic profile under the SaaS WO.

Analyzing the charts shown in Figure 11, for NC1W7WOIE, most of the time there are bursts of data being exchanged very frequently, with the majority of the said bursts being smaller 100 B in downstream (Rx) direction. The volume of data in downstream (Rx, Figure 11) is clearly smaller than the volume of data in upstream (Tx, Figure 11), except for a few particular events: (a) at the beginning of the examined trace, the volume of data in downstream is clearly larger than the volume of data in upstream, and (b) there is also a clear data reception spike around the middle of the trace. In the remainder of the trace, the volume of data in upstream is much higher (logically related with the input of data when working in the SaaS WO), something that is further confirmed in Table 6.

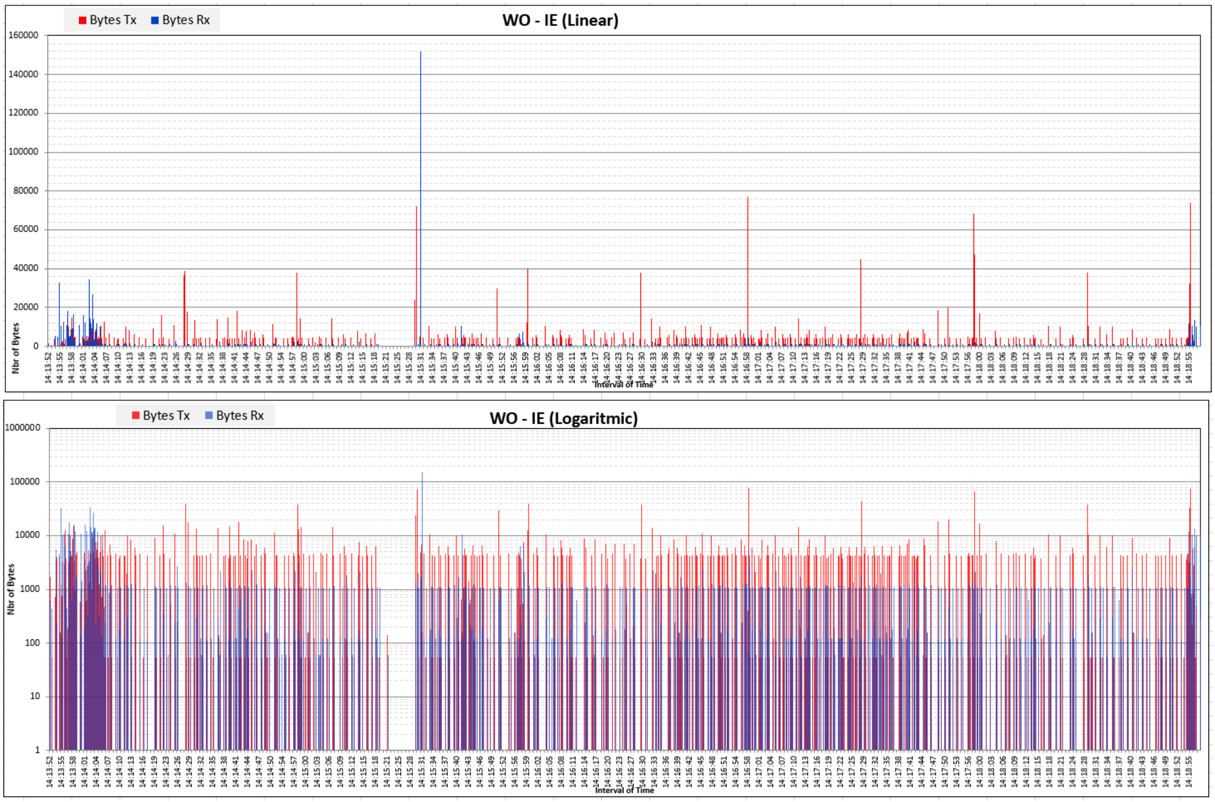


Figure 11: NC1W7WOIE - bytes exchanged in the filtered packet trace, using two scales

Table 6: NC1W7WOIE summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	1165.19	KB
Total Data Tx	2965.67	KB
Rx rate, avg	31.09	kbps
Tx rate, avg	79.14	kbps
Total (Tx + Rx) rate, avg	0.110	Mbps
Rx rate, max	1233.70	kbps
Tx rate, max	1044.79	kbps

Data Volume Ratio (Rx/Tx)
0.39

Ratio of Avg Data rate (Rx/Tx)
0.39

Ratio of Max Data rate (Rx/Tx)
1.18

For the three scenarios, NC1W7WOIE, NC1W7WOFF, and NC1W7WOGC, the volume of received and transmitted traffic volume is substantially smaller than 1 (see Table 6, Table 7, and Table 8, with Rx/Tx traffic volume ratio of 0.39, 0.26 and, 0.4, respectively). Contrary to the NC1W7GD scenarios described in the beginning of this section, this particular SaaS seems to consume much more bandwidth in the upstream direction (Tx), effectively requiring higher capacity uplink – something that is not typically available in most common residential class deployments. It is also interesting to observe that all three examined scenarios perform similarly in terms of downstream data volume (~1100 KB of data received), while NC1W7WOGC and NC1W7WOIE scenarios also perform similar in terms of data transmitted (~2900 KB). NC1W7WOFF seems to be transmitting much more information, reaching 3900 KB, almost 1000 KB more than the other two examined browsers.

Table 7: NC1W7WOFF summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	1017.33	KB
Total Data Tx	3914.37	KB
Rx rate, avg	25.25	kbps
Tx rate, avg	97.17	kbps
Total (Tx + Rx) rate, avg	0.122	Mbps
Rx rate, max	1298.03	kbps
Tx rate, max	1071.71	kbps

Data Volume Ratio (Rx/Tx)
0.26

Ratio of Avg Data rate (Rx/Tx)
0.26

Ratio of Max Data rate (Rx/Tx)
1.21

Table 8: NC1W7WOGC summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	1174.70	KB
Total Data Tx	2932.79	KB
Rx rate, avg	31.55	kbps
Tx rate, avg	78.77	kbps
Total (Tx + Rx) rate, avg	0.110	Mbps
Rx rate, max	1689.29	kbps
Tx rate, max	958.06	kbps

Data Volume Ratio (Rx/Tx)
0.40

Ratio of Avg Data rate (Rx/Tx)
0.40

Ratio of Max Data rate (Rx/Tx)
1.76

In what concerns the event correlation with the network activity, as an example we use NC1W7WOFF, since once again all studied scenarios present similar visual traffic profile. Observing the network activity correlation shown in Figure 12, it is clear that the processes of file creation, naming, and loading and closing the WO GUI in the browser represent one of the biggest data transfer, again, only comparable to the process of inserting an image, document refresh associated with the image insertion, and table formatting for all examined browsers.

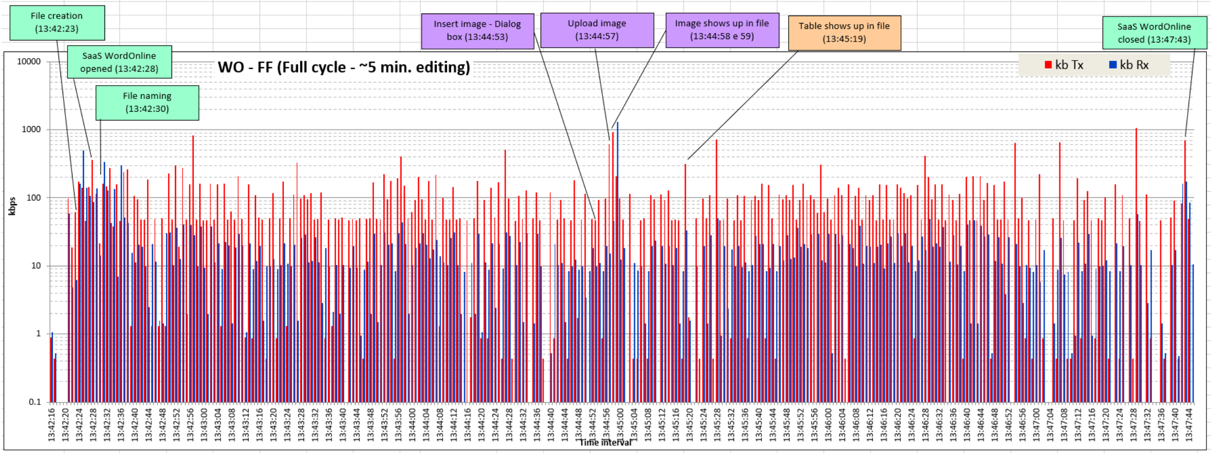


Figure 12: Correlation between throughput and chronological order of major events (in logarithmic scale) for NC1W7WOFF

Observing the chart shown in Figure 12, it is noticeable a number of upstream (Tx) direction data peaks, something that could not be correlated to any specific activity in the recorded video sessions (as shown in Figure 13). This observation is equally applicable to all three examined scenarios, i.e., NC1W7WOFF, NC1W7WOGC, and NC1W7WOIE.

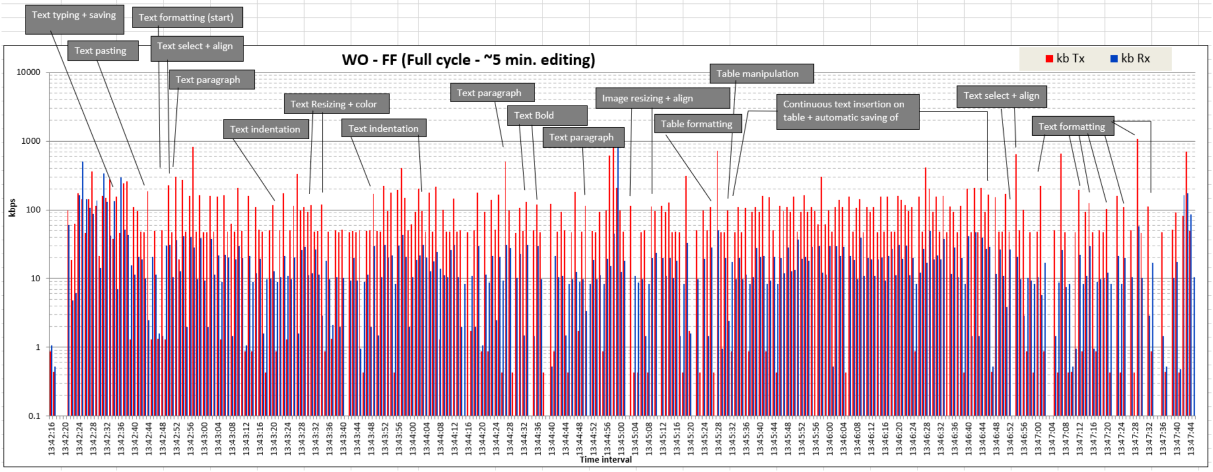


Figure 13: Correlation between throughput and chronological order of minor events (in logarithmic scale) for NC1W7WOFF

There are mainly two reasons that are certainly plausible to explain this behavior, (a) menu interaction to perform action of simple formatting of text or objects, eventually with some delay/latency or (b) background active auto-saving. The first reason not always justifies the observed network activity peaks, since at times these observed activity peaks seem to coincide with the idle time in the examined data trace (no interaction with any elements of the WO GUI, text, or document content), which again could be explained through latency of the application or the network or both.

4.1.2 NC1W7GS and NC1W7PO scenarios

This section examines the operation of SaaS Google Slides (GS) and Microsoft Powerpoint Online (PO) for reference browsers, i.e., GC, FF, and IE. The analysis for the collected data follows the process discussed in detail in section 4.1.1, with the following discussion focusing only on more interesting and critical observations distinguishing this SaaS from the previously examined text editing application.

In these scenarios, a user creates and edits a multimedia presentation, during approximately 7 minutes, with the resulting size of approximately 1 MB (for reference, when saved locally using Microsoft Office PowerPoint in pptx format), executing the following basic tasks:

- creating and naming a new multimedia presentation;
- formatting the layout and design of slide and the whole presentation;
- inserting, editing and formatting text;
- inserting and manipulating objects such as *wordArt*, images and video;
- applying animation effects to the objects and transition effects between slides; and
- previewing the presentation in the cloud.

Table 9, Table 10, and Table 11 present summary trace information for NC1W7GSFF, NC1W7GSGC, and NC1W7GSIE scenarios, respectively.

Table 9: NC1W7GSFF summary of the statistical parameters from the trace

Traffic statistics				Data Volume Ratio (Rx/Tx)
Total Data Rx	12586.25	KB		11.63
Total Data Tx	1082.13	KB		
Rx rate, avg	229.64	kbps		Ratio of Avg Data rate (Rx/Tx)
Tx rate, avg	19.74	kbps		11.63
Total (Tx + Rx) rate, avg	0.249	Mbps		
Rx rate, max	21750.55	kbps		Ratio of Max Data rate (Rx/Tx)
Tx rate, max	777.24	kbps		27.98

Table 10: NC1W7GSGC summary of the statistical parameters from the trace

Traffic statistics				Data Volume Ratio (Rx/Tx)
Total Data Rx	17528.39	KB		12.70
Total Data Tx	1380.24	KB		
Rx rate, avg	356.31	kbps		Ratio of Avg Data rate (Rx/Tx)
Tx rate, avg	28.06	kbps		12.70
Total (Tx + Rx) rate, avg	0.384	Mbps		
Rx rate, max	32242.02	kbps		Ratio of Max Data rate (Rx/Tx)
Tx rate, max	782.07	kbps		41.23

Table 11: NC1W7GSIE summary of the statistical parameters from the trace

Traffic statistics				Data Volume Ratio (Rx/Tx)
Total Data Rx	14668.79	KB		9.37
Total Data Tx	1565.26	KB		
Rx rate, avg	290.26	kbps		Ratio of Avg Data rate (Rx/Tx)
Tx rate, avg	30.97	kbps		9.37
Total (Tx + Rx) rate, avg	0.321	Mbps		
Rx rate, max	6965.20	kbps		Ratio of Max Data rate (Rx/Tx)
Tx rate, max	780.24	kbps		8.93

As expected for this type of multimedia-rich application, the data volume observed is the substantially higher (when compared with text editing SaaS) specially the downstream volume of data (12MB, 17MB, and 14MB, respectively), resulting in a very high downstream-to-upstream traffic ratio. Effectively, this SaaS is downstream-heavy and requires properly dimensioned access network to operate correctly. Most of the downstream traffic in this case is related with the playback of an embedded video, as well as image editing. Furthermore, it is also worth taking note of the relatively high maximum downstream data rate: ~22Mbps, 32Mbps, and ~7Mbps, respectively, for individual scenarios, clearly indicating the high-speed data download required for proper video and image display, perhaps covering also the case of local caching (especially in the case of GC interacting with this SaaS).

Similar observations can be also made for NC1W7POFF, NC1W7POGC, and NC1W7POIE scenarios, summarized in Table 12, Table 13, and Table 14, respectively. It is worth noting, though, that this scenario data volume (both in upstream and downstream) is the much higher. Specifically, the volume of the upstream traffic is at least 4-5 times the traffic volume for SaaS GS, implying much higher volume of data transmitted towards the SaaS PO host(s) and at a much higher rate.

These observations are further substantiated by looking at the correlation between individual actions and the resulting network activity for specific scenarios, i.e.: NC1W7GSFF and NC1W7GSIE scenarios in Figure 14 and NC1W7POFF and NC1W7POIE scenarios in Figure 15. Charts for GC are not presented for both SaaS scenarios due to their visual similarity to respective FF charts.

Table 12: NC1W7POFF summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	18025.41	KB
Total Data Tx	8079.17	KB
Rx rate, avg	336.36	kbps
Tx rate, avg	150.76	kbps
Total (Tx + Rx) rate, avg	0.487	Mbps
Rx rate, max	26288.02	kbps
Tx rate, max	1503.21	kbps

Data Volume Ratio (Rx/Tx)	Ratio of Avg Data rate (Rx/Tx)	Ratio of Max Data rate (Rx/Tx)
2.23	2.23	17.49

Table 13: NC1W7POGC summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	21649.32	KB
Total Data Tx	8482.15	KB
Rx rate, avg	393.24	kbps
Tx rate, avg	154.07	kbps
Total (Tx + Rx) rate, avg	0.547	Mbps
Rx rate, max	20874.44	kbps
Tx rate, max	1524.46	kbps

Data Volume Ratio (Rx/Tx)	Ratio of Avg Data rate (Rx/Tx)	Ratio of Max Data rate (Rx/Tx)
2.55	2.55	13.69

Table 14: NC1W7POIE summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	19181.93	KB
Total Data Tx	7690.89	KB
Rx rate, avg	362.91	kbps
Tx rate, avg	145.51	kbps
Total (Tx + Rx) rate, avg	0.508	Mbps
Rx rate, max	8921.93	kbps
Tx rate, max	1422.13	kbps

Data Volume Ratio (Rx/Tx)	Ratio of Avg Data rate (Rx/Tx)	Ratio of Max Data rate (Rx/Tx)
2.49	2.49	6.27

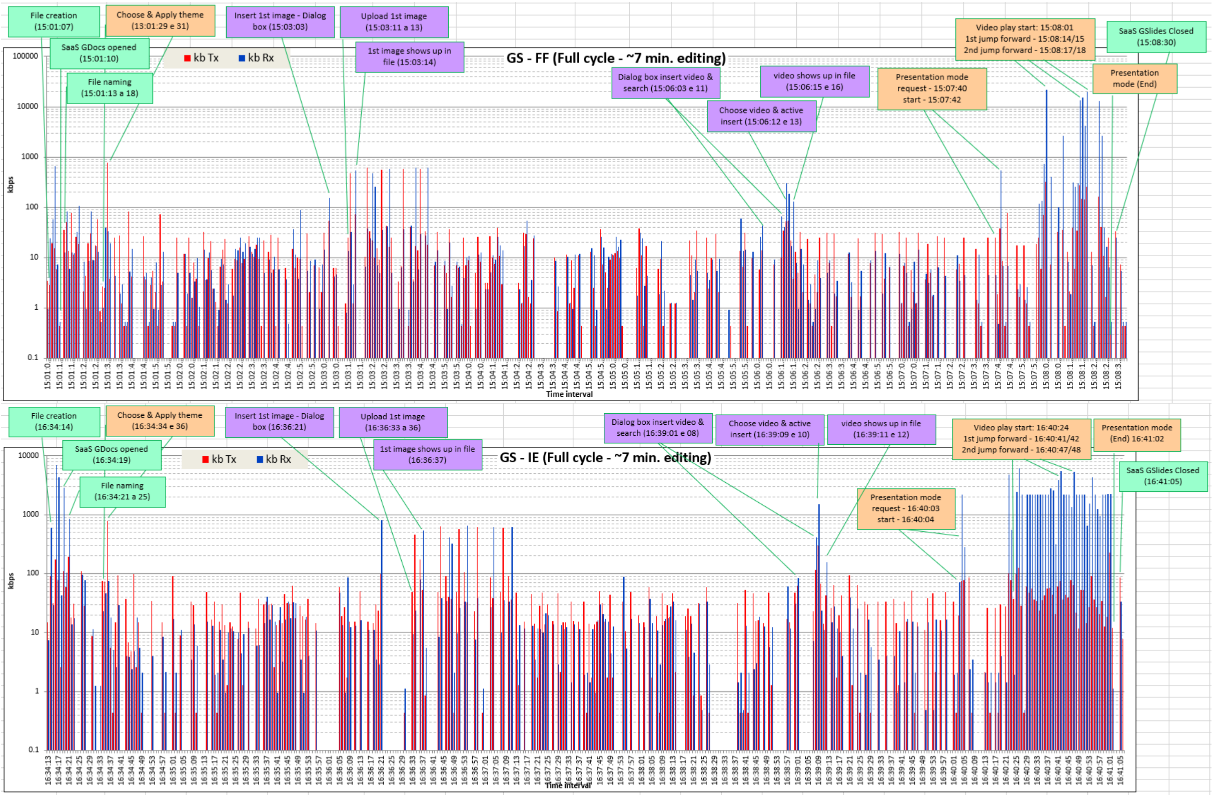


Figure 14: Correlation between throughput and chronological order of major events (in logarithmic scale) for NC1W7GSFF and NC1W7GSIE (top to bottom)

Comparing these activity charts with similar charts produced for text editing SaaS scenarios, it is immediately clear that the amount and intensity of data exchange is much higher in this case. Most of the major events can be also clearly correlated to network activity peaks, though a number of them remains unlabeled – they are related to the insertion of slides and text, as well as text editing and formatting activities or inclusion of animations and transitions between individual slides. Comparing the presentation and text editing SaaS scenarios, the conclusions are similar, i.e., any text editing actions on their own are not network traffic intensive, though the sheer volume of text formatting changes may have some impact on background network traffic. This may prove important especially in the case of congested or bandwidth-constrained networks, where such background traffic may increase congestion and cause SaaS QoE degradation.

It is also possible to observe that the NC1W7GSIE is more network intensive (when compared with NC1W7GSFF and NC1W7GSGC) when loading the GUI interface (at the beginning of the scenario) and when running in the presentation mode (at the end of the scenario). In the case of all examined browsers for both SaaS applications, the maximum data rates can be closely correlated with the presentation preview mode, and specifically – the video replay. Similar observations can be made for individual browsers for the SaaS PO.

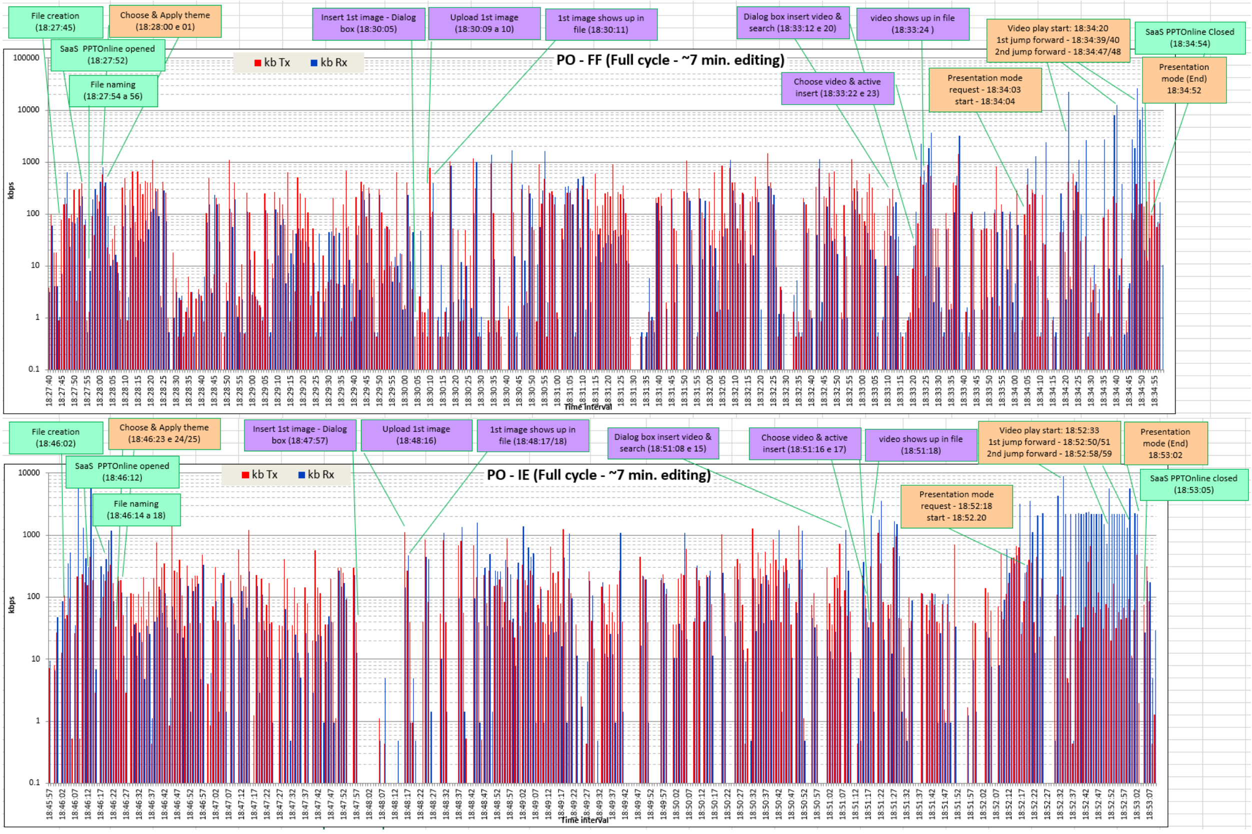


Figure 15: Correlation between throughput and chronological order of major events (in logarithmic scale) for NC1W7POFF and NC1W7POIE (top to bottom)

4.1.3 NC1W7GX and NC1W7XO scenarios

This section examines the operation of SaaS Google Sheets (GX) and Microsoft Excel Online (XO) for reference browsers, i.e., GC, FF, and IE. The analysis for the collected data follows the process discussed in detail in section 4.1.1, again, with the following discussion focusing only on more interesting and critical observations distinguishing this SaaS from the previously examined text and presentation editing applications.

In these scenarios, a user creates and edits a workbook containing two spreadsheets with the resulting size of approximately 50 KB (for reference, when saved locally using Microsoft Office Excel in xlsx format), executing the following basic tasks:

- creating and naming a new workbook;
- inserting text and formatting cells and/or groups of cells and the tables;
- using formulas and functions to manipulate data sets and obtain new relevant data;
- inserting and customizing images and charts from data in the workbook;
- copying data between spreadsheets of the workbook;
- use filters to obtain specific data from tables;

Table 15, Table 16, and Table 17 present summary trace information for NC1W7GXFF, NC1W7GXGC, and NC1W7GXIE scenarios, respectively. It is immediately visible that the spreadsheet SaaS has behavior closer to text editing SaaS than to presentation editing SaaS, at least for FF and GC, given that the volume of upstream traffic is larger than the volume of downstream traffic, implying much more data upload to the SaaS host(s). Such a behavior is expected, since spreadsheet operations rely mostly on simple data exchange, and presentation of results, rather than video and image intensive presentations. However, it is also worth noting that IE (see Table 17) is clearly an outlier in this case, showing spreadsheet SaaS statistics similar to the presentation SaaS statistics, i.e., much higher volume of downstream traffic (~10 times the upstream traffic volume), and a similar pattern for the maximum data rate observed. While both FF and GC have higher maximum upstream data rate (~3 times the downstream data rate), the IE downstream maximum data rate is 24 times higher than upstream data rate.

Table 15: NC1W7GXFF summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	477.48	KB
Total Data Tx	667.27	KB
Rx rate, avg	6.26	kbps
Tx rate, avg	8.75	kbps
Total (Tx + Rx) rate, avg	0.015	Mbps
Rx rate, max	876.14	kbps
Tx rate, max	2237.44	kbps

Data Volume Ratio (Rx/Tx)
0.72

Ratio of Avg Data rate (Rx/Tx)
0.72

Ratio of Max Data rate (Rx/Tx)
0.39

Table 16: NC1W7GXGC summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	435.73	KB
Total Data Tx	755.58	KB
Rx rate, avg	6.88	kbps
Tx rate, avg	11.93	kbps
Total (Tx + Rx) rate, avg	0.019	Mbps
Rx rate, max	885.34	kbps
Tx rate, max	2280.56	kbps

Data Volume Ratio (Rx/Tx)
0.58

Ratio of Avg Data rate (Rx/Tx)
0.58

Ratio of Max Data rate (Rx/Tx)
0.39

Table 17: NC1W7GXIE summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	11475.77	KB
Total Data Tx	1024.62	KB
Rx rate, avg	162.93	kbps
Tx rate, avg	14.55	kbps
Total (Tx + Rx) rate, avg	0.487	Mbps
Rx rate, max	29529.22	kbps
Tx rate, max	1227.92	kbps

Data Volume Ratio (Rx/Tx)
11.20

Ratio of Avg Data rate (Rx/Tx)
11.20

Ratio of Max Data rate (Rx/Tx)
24.05

Table 18, Table 19, and Table 20 present summary trace information for NC1W7XOFF, NC1W7XOGC, and NC1W7XOIE scenarios, respectively. Clearly, the volume of traffic exchanged for SaaS XO is much higher when compared with SaaS GX, and with all three browsers showing consistent behavior. For example, FF uploads ~5MB and downloads ~2MB for SaaS XO while the very same browser uploads only 667KB and downloads 477KB for SaaS GX. IE is the only exception in this case, where SaaS GX generated more downstream data (11475KB) when compared to SaaS XO (2528KB). All browsers with SaaS XO exhibit a strong upstream preference, i.e., uploading much more information to the SaaS host(s) then receiving from the SaaS.

Table 18: NC1W7XOFF summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	1943.48	KB
Total Data Tx	5227.52	KB
Rx rate, avg	40.00	kbps
Tx rate, avg	107.60	kbps
Total (Tx + Rx) rate, avg	0.148	Mbps
Rx rate, max	514.64	kbps
Tx rate, max	1336.71	kbps

Data Volume Ratio (Rx/Tx)
0.37

Ratio of Avg Data rate (Rx/Tx)
0.37

Ratio of Max Data rate (Rx/Tx)
0.39

Table 19: NC1W7XOGC summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	2871.34	KB
Total Data Tx	6254.86	KB
Rx rate, avg	52.27	kbps
Tx rate, avg	113.87	kbps
Total (Tx + Rx) rate, avg	0.166	Mbps
Rx rate, max	1577.22	kbps
Tx rate, max	1440.98	kbps

Data Volume Ratio (Rx/Tx)
0.46

Ratio of Avg Data rate (Rx/Tx)
0.46

Ratio of Max Data rate (Rx/Tx)
1.09

Table 20: NC1W7XOIE summary of the statistical parameters from the trace

Traffic statistics		
Total Data Rx	2528.97	KB
Total Data Tx	5459.78	KB
Rx rate, avg	48.75	kbps
Tx rate, avg	105.24	kbps
Total (Tx + Rx) rate, avg	0.154	Mbps
Rx rate, max	1176.05	kbps
Tx rate, max	1346.07	kbps

Data Volume Ratio (Rx/Tx)
0.46

Ratio of Avg Data rate (Rx/Tx)
0.46

Ratio of Max Data rate (Rx/Tx)
0.87

These observations are further substantiated by looking at the correlation between individual actions and the resulting network activity for specific scenarios, i.e.: NC1W7GXFF and NC1W7GXIE scenarios in

Figure 16 and NC1W7XOFF and NC1W7XOIE scenarios in Figure 17. Charts for GC are not presented for both SaaS scenarios due to their visual similarity to respective FF charts.

Comparing these activity charts with similar charts produced for text editing SaaS scenarios, it is immediately clear that the amount and intensity of data exchange is lower in this case. Most of the major events can be also clearly correlated to network activity peaks.

It is also worth noting that the SaaS GX seems to be more network intensive when compared with SaaS XO, especially in the case of IE, where the volume of data exchanged for SaaS GX is compatible with presentation editing SaaS. IE would not be therefore expected to perform very well with SaaS GX when using congested or bandwidth constrained access networks. Other browser and SaaS combinations behave much better, though SaaS XO does require access network with the upstream able to burst to at least 1-2 Mbps to provide good QoE to end users.

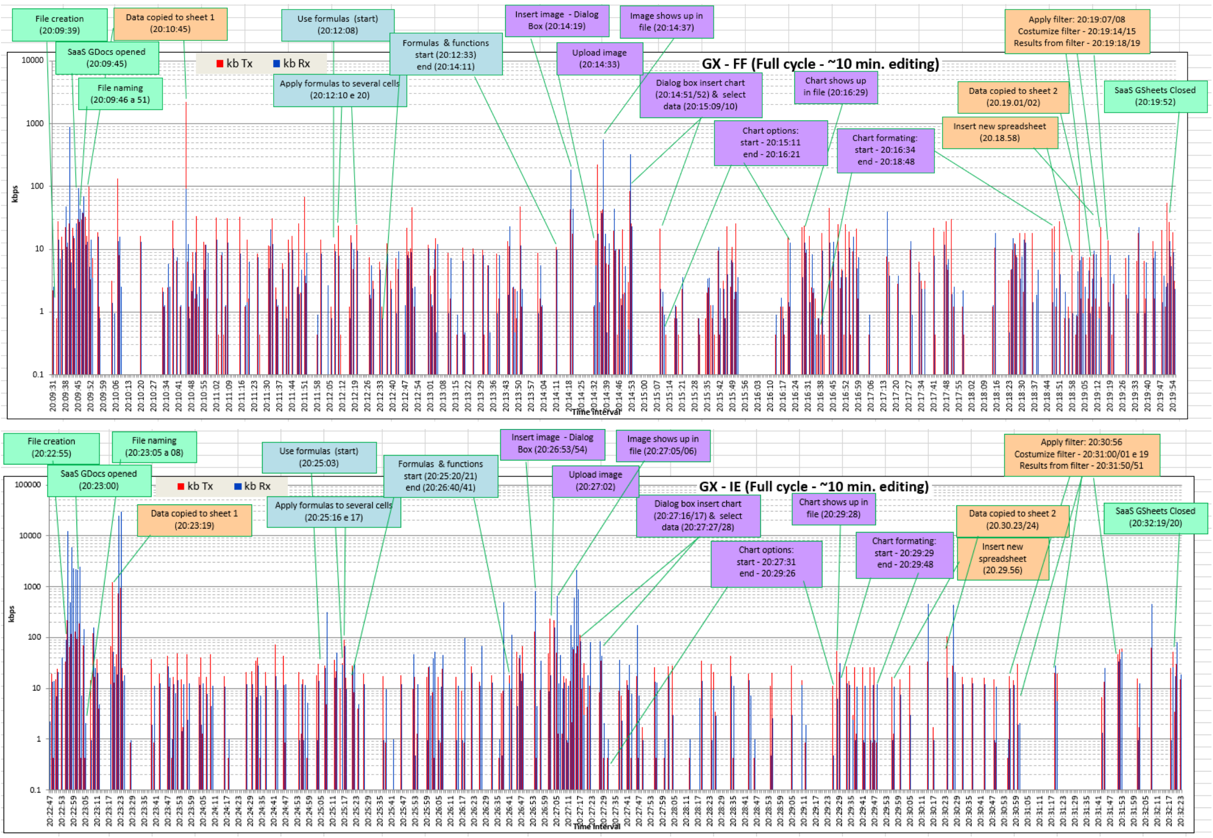


Figure 16: Correlation between throughput and chronological order of major events (in logarithmic scale) for NC1W7GXFF and NC1W7GXIE (top to bottom)

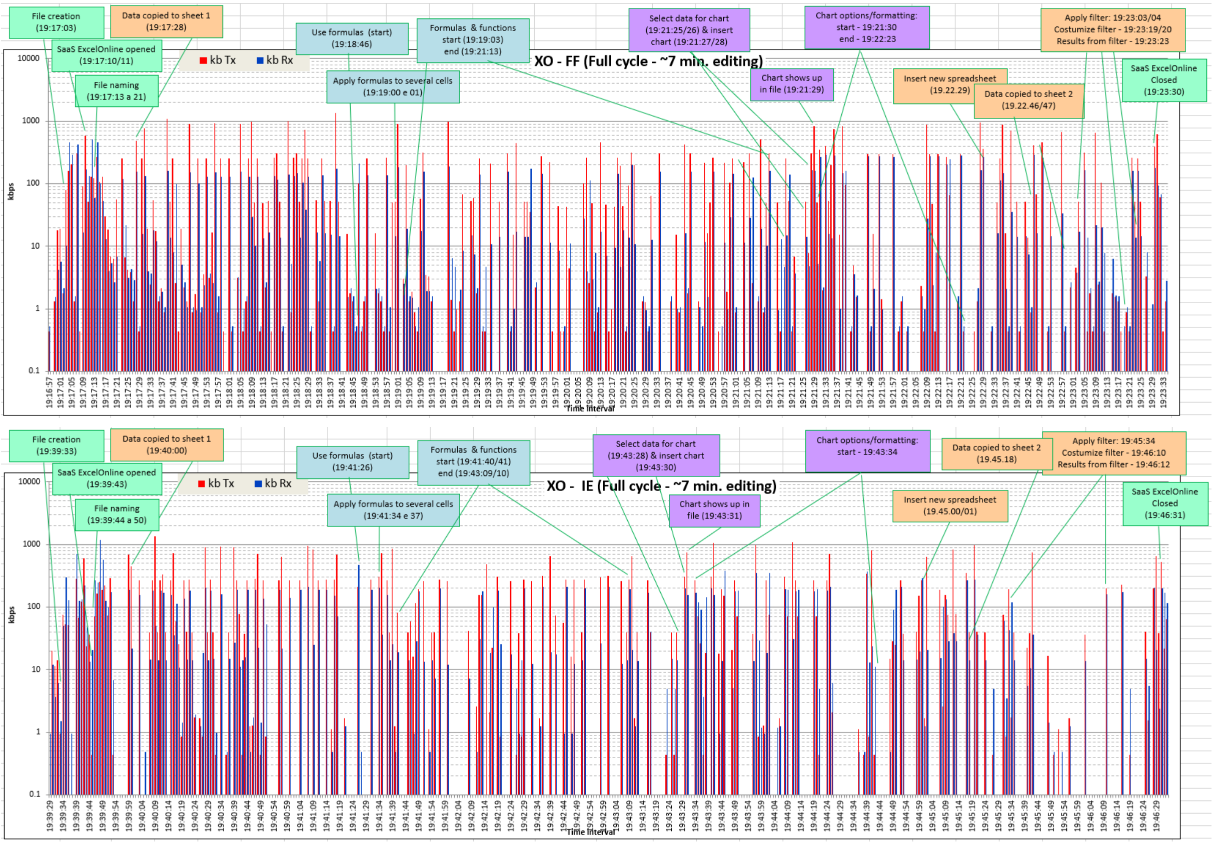


Figure 17: Correlation between throughput and chronological order of major events (in logarithmic scale) for NC1W7XOFF and NC1W7XOIE (top to bottom)

4.1.4 Summary of results for Hypothesis A

Looking at the individual SaaS application and browser combinations presented in previous sections, it is clearly possible to correlate specific SaaS application-level actions (text insertion and editing, image insertion, inclusion of other multimedia objects and formulas, when applicable ,etc.) with the network-level activity peaks. The correlation becomes much less obvious for non-major events (text typing, paragraph insertion, text and table formatting, etc.) which do not require exchange of a lot of information between the end-user client and SaaS host(s).

It is worth also noting inconsistency in browser behavior for the same SaaS in some cases. It seems that both FF and GC perform similarly in most cases, while IE remains an outlier in terms of performance, traffic profile, etc. This may be related with the similar underlying implementation in FF and GC, similar rendering engines, and likely – much more modern design when compared with aging IE. The additional optimizations for Google-based SaaS applications and GC are further explored in detail in the following hypotheses, and should be a topic of a future separate study, especially taking into account next generation Microsoft browsers and their potential optimization for the use with Microsoft provided SaaS applications.

4.2 Hypothesis B: SaaS is delivered to end user over TCP

Any SaaS application provided to end users across the network requires some reliability guarantees, especially in terms of saving document content and retrieving the content saved in the SaaS. From the existing transport layer (L4) protocols (UDP and TCP), only TCP provides connection-oriented operation, with data error detection and recovery capabilities. UDP is generally classified as a best-effort transport layer protocol, providing no delivery guarantee, no error detection, and no error recovery capability. From the two commonly available transport layer protocols, the TCP is the best candidate for the SaaS implementation.

This hypothesis was examined by verifying the types of L4 protocols in use for the individual examined scenarios, as summarized in the following tables.

- NC1: Table 21 for GD and Table 22 for WO,
- NC2: Table 23 for GD and Table 24 for WO,
- NC3: Table 25 for GD and Table 26 for WO.

Note that a) for NC1, due to an initial different set of goals, the SaaS was only analyzed under W7 Pro and b) IE is not available on KU, hence the respective cells are marked as “N/A”.

Looking at the aforementioned tables, Hypothesis B holds for all scenarios excluding any GD applications when accessed using GC. GC relies on QUIC (Google Quick UDP Internet Connections, for details of this still experimental protocol please see https://datatracker.ietf.org/wg/quic/charter/) for data exchange between SaaS hosts and the end user client. QUIC operates using UDP transport layer protocol [37], while data error detection and recovery (including retransmission) are relegated to the application layer (L7) and implemented both in GC as well as the SaaS hosts. There is still some background volume of TCP traffic being exchanged as well, though its volume is substantially smaller when compared with the UDP traffic.

Obviously, when browsers other than GC access GD applications, the SaaS hosts are capable of providing services over standard TCP connections, as demonstrated in Table 21, Table 23, and Table 25 for GC, FF, and IE across different operating systems for NC1, NC2, and NC3, respectively. The observations apply to both IPv4 (NC1 and NC2) as well as IPv6 (NC3).

Similarly, GC is capable of using TCP connections when accessing other SaaS providers, as shown in Table 22, Table 24, and Table 26 in different network configurations, where it is used to access WO. In this respect, this particular browser is no different than FF or IE when running under any of the examined operating systems. The observations apply to both IPv4 (NC1 and NC2) as well as IPv6 (NC3).

Table 21: L4 protocol types, GD scenarios, NC1*

Browser / OS	GC	FF	IE
W7	TCP (primary) UDP/QUIC (secondary)	TCP	TCP

(*) Based in the analysis of one collected trace (2016)

Table 22: L4 protocol types, WO scenarios, NC1*

Browser / OS	GC	FF	IE
W7	TCP	TCP	TCP

(*) Based in the analysis of one collected trace (2016)

Table 23: L4 protocol types, GD scenarios, NC2

Browser / OS	GC	FF	IE
W7	UDP/QUIC (primary) TCP (secondary)	TCP	TCP
W10			
KU			N/A

Table 24: L4 protocol types, WO scenarios, NC2

Browser / OS	GC	FF	IE
W7	TCP	TCP	TCP
W10			
KU			N/A

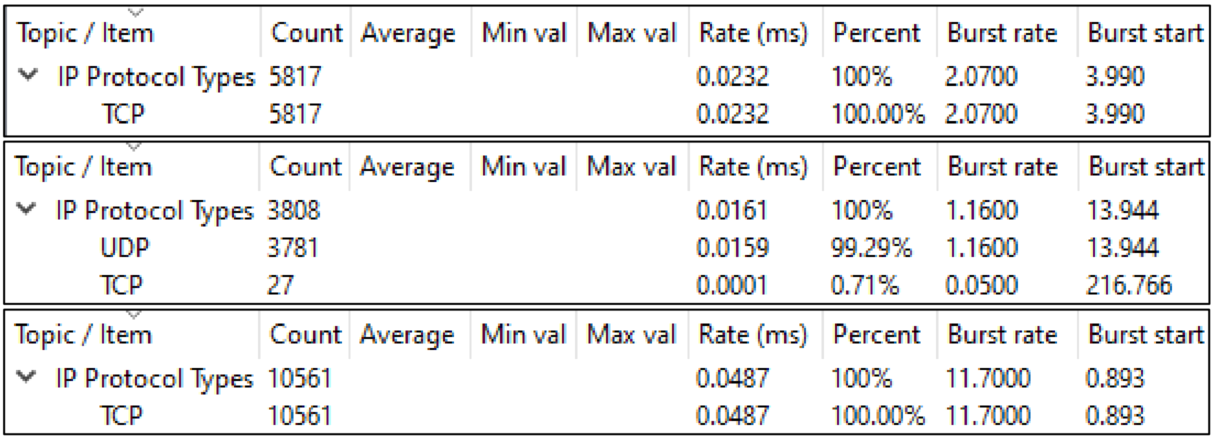
Table 25: L4 protocol types, GD scenarios, NC3

Browser / OS	GC	FF	IE
W7	UDP/QUIC (primary) TCP (secondary)	TCP	TCP
W10			
KU			N/A

Table 26: L4 protocol types, WO scenarios, NC3

Browser / OS	GC	FF	IE
W7	TCP	TCP	TCP
W10			
KU			N/A

As an example, specific values for two different operating systems (W7 and KU) are demonstrated in detail for different browsers, providing an example of raw data used to generate individual summary tables. Figure 18 presents the protocol type and packet counts for GC, FF, and IE for W7 under NC2 and Figure 19 presents the protocol type and packet counts for GC and FF for KU under NC2 (from Wireshark).

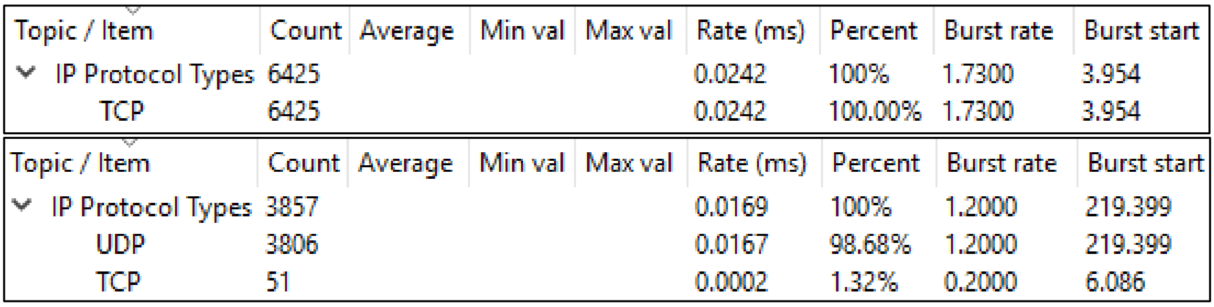


Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
⌄ IP Protocol Types	5817				0.0232	100%	2.0700	3.990
TCP	5817				0.0232	100.00%	2.0700	3.990

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
⌄ IP Protocol Types	3808				0.0161	100%	1.1600	13.944
UDP	3781				0.0159	99.29%	1.1600	13.944
TCP	27				0.0001	0.71%	0.0500	216.766

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
⌄ IP Protocol Types	10561				0.0487	100%	11.7000	0.893
TCP	10561				0.0487	100.00%	11.7000	0.893

Figure 18: NC2W7GDFF, NC2W7GDGC, and NC2W7GDIE protocol statistics (from top to bottom)

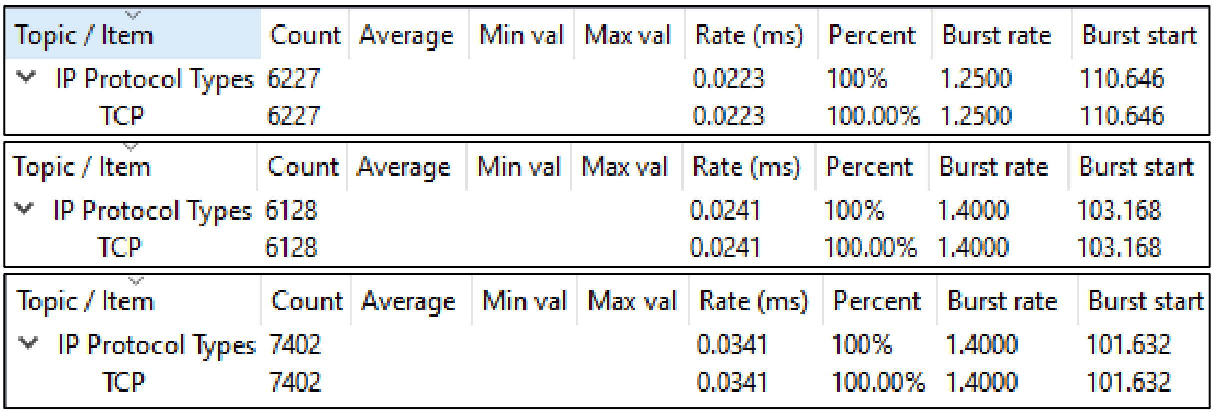


Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
⌄ IP Protocol Types	6425				0.0242	100%	1.7300	3.954
TCP	6425				0.0242	100.00%	1.7300	3.954

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
⌄ IP Protocol Types	3857				0.0169	100%	1.2000	219.399
UDP	3806				0.0167	98.68%	1.2000	219.399
TCP	51				0.0002	1.32%	0.2000	6.086

Figure 19: NC2KUGDFF and NC2KUGDGC protocol statistics (from top to bottom)

Figure 20 shows an example of protocol type and packet counts for GC, FF, and IE for W10 under NC2 when using WO.



Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
⌄ IP Protocol Types	6227				0.0223	100%	1.2500	110.646
TCP	6227				0.0223	100.00%	1.2500	110.646

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
⌄ IP Protocol Types	6128				0.0241	100%	1.4000	103.168
TCP	6128				0.0241	100.00%	1.4000	103.168

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
⌄ IP Protocol Types	7402				0.0341	100%	1.4000	101.632
TCP	7402				0.0341	100.00%	1.4000	101.632

Figure 20: NC2W10WOFF, NC2W10WOGC, and NC2W10WOIE protocol statistics (from top to bottom)

4.3 Hypothesis C: SaaS generates more upstream than downstream traffic

Given the nature of SaaS applications (user data is stored and processed in the cloud), it is expected that the volume of upstream traffic (data generated by the end user and uploaded to the SaaS cloud) is greater than the volume of downstream traffic (any updates transmitted by the SaaS to the end user client). Furthermore, it is expected that the local end user client is capable of displaying information on its own, only periodically synchronizing status with the SaaS host, further emphasizing the expected lower volume of downstream traffic. Note that the majority of typical Internet applications are download-only, where the end user consumes data stored on the server (for example, watching a video online, listening to music streamed to the local player, etc.), generating very little (if any) upstream traffic, mostly in the form of control traffic.

This hypothesis was examined by verifying the volume of upstream and downstream traffic exchanged between the SaaS hosts and the end user client for the individual examined scenarios, as summarized in the following tables.

- NC1: Table 29 for GD and Table 30 for WO;
- NC2: Table 31 for GD and Table 32 for WO;
- NC3: Table 33 for GD and Table 34 for WO.

Note that IE is not available on KU, hence the respective cells are marked as "N/A". It is worth mentioning also that individual traces collected for each examined scenario have been normalized, as discussed in section 3.5, effectively providing analysis based on average packet trace for the given scenario.

Table 27 and Table 28 show upstream and downstream traffic volume (respectively) for the average packet trace profile for the NC2KUGDFF scenario, with the total traffic volume in each referenced table marked highlighted in yellow. The traffic ratio is then obtained by dividing the aggregate upstream and downstream traffic volume and recorded in the respective summary table (in this case, Table 31).

Table 27: Scenario NC2KUGDFF, upstream traffic volume per SaaS host number

Average	Samples (B)								Host
(B)	#1	#	#3	#4	#5	#6	#7	#8	#
632928	679591	630476	684405	627762	598806	602306	615420	624653	**1**
463192	455440	457644	458396	537502	429754	297959	529491	539348	**2**
76568	30022	71177	42114	74128	78485	219606	64539	32469	**3**
27597	11377	27249	14035	26850	73669	36785	14884	15923	**4**
12540	8772	19183	12080	4921	14038	17365	13395	10564	**5**
6429	6135	10889	5105	4181	7065	6244	6410	5404	**6**
5174	5196	5696	5039	4102	6775	4182	5732	4667	**7**
4453	4180	4181	4183		5242	4147	5041	4195	**8**
3433	2166	4148	4147		4182		4216	1737	**9**
2243			2058		2389		4094	432	**10**
347			261		432				**11**
1206 KB									**12**

Table 28: Scenario NC2KUGDFF, downstream traffic volume per SaaS host number

Average (B)	Samples (B)								Host
	#1	#	#3	#4	#5	#6	#7	#8	#
632928	610860	583147	607825	687089	601742	406621	627839	572375	**1**
463192	234450	284566	452309	436984	565413	254052	574594	228228	**2**
76568	133432	178138	229927	134320	81576	75777	234186	119691	**3**
27597	78027	87802	78957	69639	77855	55994	108130	74445	**4**
12540	76294	80156	75929	30444	37385	54100	95403	18307	**5**
6429	7799	33281	11591	5691	14348	18494	20734	14872	**6**
5174	6346	22827	9737	3449	13113	3467	12590	6490	**7**
4453	3276	3616	6386		10158	3132	6286	6313	**8**
3433	1751	3202	3695		8133		3285	1382	**9**
2243			3274		3482		3272	237	**10**
347			132		237				**11**
1259 KB									**12**

Up to two decimal places are shown in individual ratios. Individual traffic volumes are expressed in units of KB (where 1KB = 1024B). Again, in order to maintain document readability, only summary tables for other examined scenarios are presented, along with the resulting conclusions.

Continuing with the example presented above, looking at Table 31 for NC2, it is clear that Hypothesis C holds for all scenarios excluding the case of FF running under KU, where the volume of downstream traffic is ~5% larger than the volume of upstream traffic. When running under W7 or W10, the very same browser meets the Hypothesis C expectations, generating 1.07 and 1.6 times more upstream traffic than downstream traffic, respectively. GC seems to perform more consistently under W7 and W10, generating almost 3 times more upstream traffic than downstream traffic. Similarly, looking at Table 32 for NC2 and covering the WO scenarios across all examined browsers and operating systems, Hypothesis C holds as well, with the upstream to downstream traffic ratio ranging from 1.61 (IE under W10) to 4.14 (GC under KU). Again, GC seems to exhibit the higher traffic ratio (4 and above), with FF in second place (ranging from 3.21 to 3.54) and IE last (1.61 and 2.85, providing the least consistent results).

Table 29: Traffic volume and upstream / downstream ratio, GD scenarios, NC1*

Browser / OS	GC	FF	IE
W7	556KB / 504KB 1.1	480KB / 449KB 1.07	2188KB / 729KB 3

(*) Based in the analysis of one collected trace (2016)

Table 30: Traffic volume and upstream / downstream ratio, WO scenarios, NC1*

Browser OS	GC	FF	IE
W7	3003KB / 1202KB 2.5	4008KB / 1041KB 3.85	3036KB / 1193KB 2.55

(*) Based in the analysis of one collected trace (2016)

Table 31: Traffic volume and upstream / downstream ratio, GD scenarios, NC2

Browser OS	GC	FF	IE
W7	1221KB / 417KB 2.93	1170KB / 1094KB 1.07	3832KB / 1825KB 2.1
W10	1397KB / 467KB 2.99	1212KB / 756KB 1.6	1457KB / 635KB 2.29
KU	1315KB / 566KB 2.32	1206KB / 1259KB 0.96	N/A

Table 32: Traffic volume and upstream / downstream ratio, WO scenarios, NC2

Browser OS	GC	FF	IE
W7	3452KB / 849KB 4.06	3455KB / 1076KB 3.21	3786KB / 1329KB 2.85
W10	3329KB / 833KB 4	3002KB / 848KB 3.54	2936KB / 1829KB 1.61
KU	2716KB / 656KB 4.14	2932KB / 779KB 3.76	N/A

Table 33: Traffic volume and upstream / downstream ratio, GD scenarios, NC3

Browser OS	GC	FF	IE
W7	1536KB/462KB 3.32	1418KB / 913KB 1.55	2128kB / 3641kB 0.58
W10	1269KB / 527KB 2.41	1537KB / 609KB 2.52	1577KB / 677KB 2.33
KU	1328KB / 489KB 2.72	1322KB / 1023KB 1.29	N/A

Table 34: Traffic volume and upstream / downstream ratio, WO scenarios, NC3

Browser OS	GC	FF	IE
W7	2677KB / 524KB 5.1	2723KB / 512KB 5.32	2889KB / 1113KB 2.6
W10	3788KB / 986KB 3.84	3362KB / 856KB 3.93	4204KB / 1246KB 3.37
KU	3253KB / 841KB 3.87	3805KB / 1084KB 3.51	N/A

Looking at data summarized in the above tables, it is possible to conclude that FF and GC, despite small variations, exhibit similar behavior with GD. It is also possible to conclude that IE, although developed by Microsoft, does not seem to be performing very well with WO, the SaaS developed by the very same company. The age of this particular browser as well as its end of life status might be one of the possible explanations of this status.

Based on these results, it is also expected that in the case of upstream bandwidth constrained networks (e.g., mobile network, satellite networks), more network intensive applications such as GS or PO might suffer from performance degradation due to the upstream network congestion.

4.4 Hypothesis D: SaaS behavior is operating system independent

Given the pervasive character of SaaS and the support of the said cloud-based application across a variety of operating systems, it is expected that the selected SaaS application will exhibit the same behavior for one selected browser across different operating systems. After all, the application is running in the cloud and the user-accessible GUI is displayed in a browser window, something that requires no execution of any end-client-side application specific to the given SaaS.

To validate this thesis, the selected SaaS application (GD) was examined using the selected browser (FF) for three different operating systems, i.e., W7, W10, and KU. FF was selected due to its pervasive and cross platform character. GC was excluded to eliminate the use of QUIC and force the SaaS host and end-user client to communicate over the more pervasive TCP/IP transport protocol.

The latest available version of the browser for the given operating system at the time of the study was used, to make sure that all the latest updates and patches are in place, potentially impacting the performance of the given browser the same way. Each operating system was fully updated at the time of the study for the very same reason. Please note that W7 lost the official Microsoft support as of January 14, 2020. This hypothesis was validated in one network configuration scenario only (NC2).

Each scenario had at least seven separate traces collected and then averaged, using the methodology described in section 3.4. The resulting average traces for each examined browser were then normalized to 0 – 100% range for packet and byte count, to provide means for normalized comparison between individual averaged traces. Finally, individual normalized averaged traces are then compared, by subtracting the observed packet and byte counts (normalized) for individual hosts and plotting the resulting data, as shown in:

- Figure 21 for comparison of SaaS GD, under FF in NC2, under W7 and 10,
- Figure 22 for comparison of SaaS GD under FF in NC2, under W7 and KU,
- Figure 23 for comparison of SaaS GD under FF in NC2, under W10 and KU.

The hypothesis indicated that the behavior for the very same SaaS under the same browser under different operating systems is the same, i.e., the normalized packet and byte counts were expected to be similar (with the assumed tolerance level below 1%). Any deviations from this value (differences above 1%) are indicative of a different behavior of the examined SaaS between the two studied browsers.

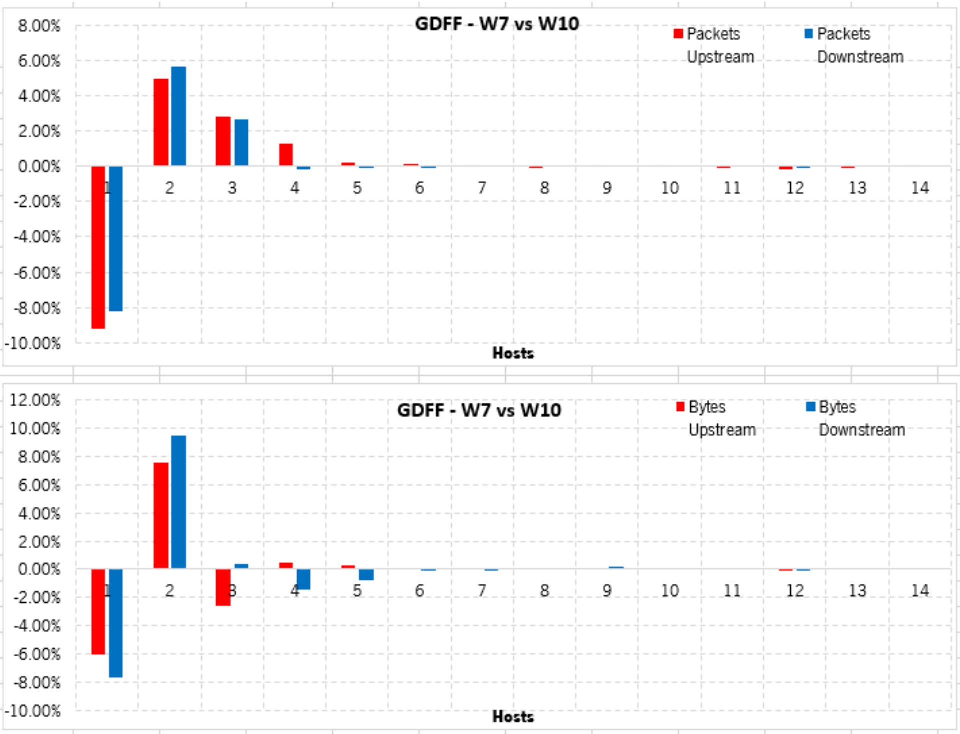


Figure 21: NC2W7GDFF versus NC2W10GDFF (normalized values) (Packets – top; Bytes – bottom)

Looking at Figure 21 (top) and comparing the behavior of the very same SaaS under the same browser in the same network (NC2), it is immediately visible that the observed behavior is substantially different between two examined operating systems. The primary host (host generating the largest amount of data) generates ~9% more packets under W10 (negative value is observed), while hosts number 2, 3, and 4 generate more packets under W7 (positive values are observed). Any differences for higher order hosts (number 5 and higher) can be safely disregarded, given the differences well below 1%.

The bottom portion of Figure 21 shows the volume of data (bytes) per host and again – W10 exchanges more information with the primary SaaS host (host number 1, observed value is negative) while the secondary SaaS host (host number 2,observed value is positive) exchanges more data under W7. Hosts number 3 and 4 are split between upstream and downstream directions; where in host number 3 the upstream direction is skewed towards W10 (negative value) and downstream is marginally skewed towards W7 (small positive value), whereas host number 4 shows the inverse behavior. Any differences for higher order hosts (number 5 and higher) can be safely disregarded, given the differences well below 1%.

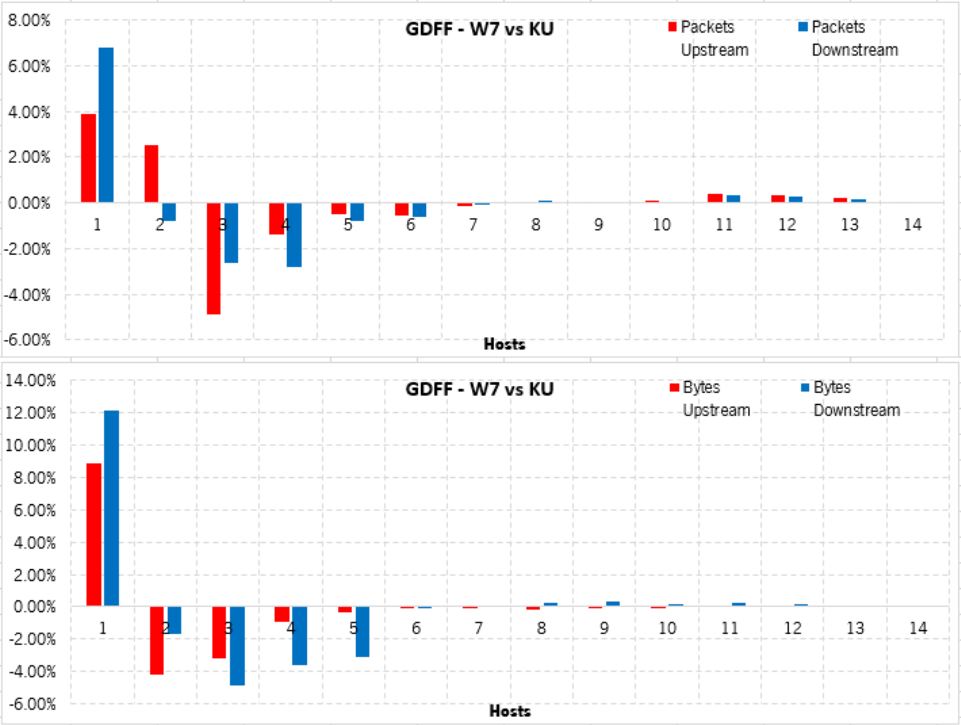


Figure 22: NC2W7GDFF versus NC2KUGDFF (normalized values) (Packets – top; Bytes – bottom)

The top portion of Figure 22 compares the number of packets (normalized) exchanged under W7 and KU, with positive values indicating higher number of packets observed under W7. Host number 1 generates more packets under W7 (positive values for upstream and downstream directions), while host 2 is split between W7 (upstream direction) and KU (downstream direction, close to 2% difference). Hosts 3 through 6 are clearly generating more packets under KU, with hosts 5 and 6 barely reaching the 1% cutoff threshold. Any differences for higher order hosts (number 7 and higher) can be safely disregarded, given the differences well below 1%.

The bottom portion of Figure 22 compares the number of bytes (normalized) exchanged under W7 and KU, with positive values indicating higher number of bytes observed under W7. Observations made for the packet count are similarly applicable to the byte count.

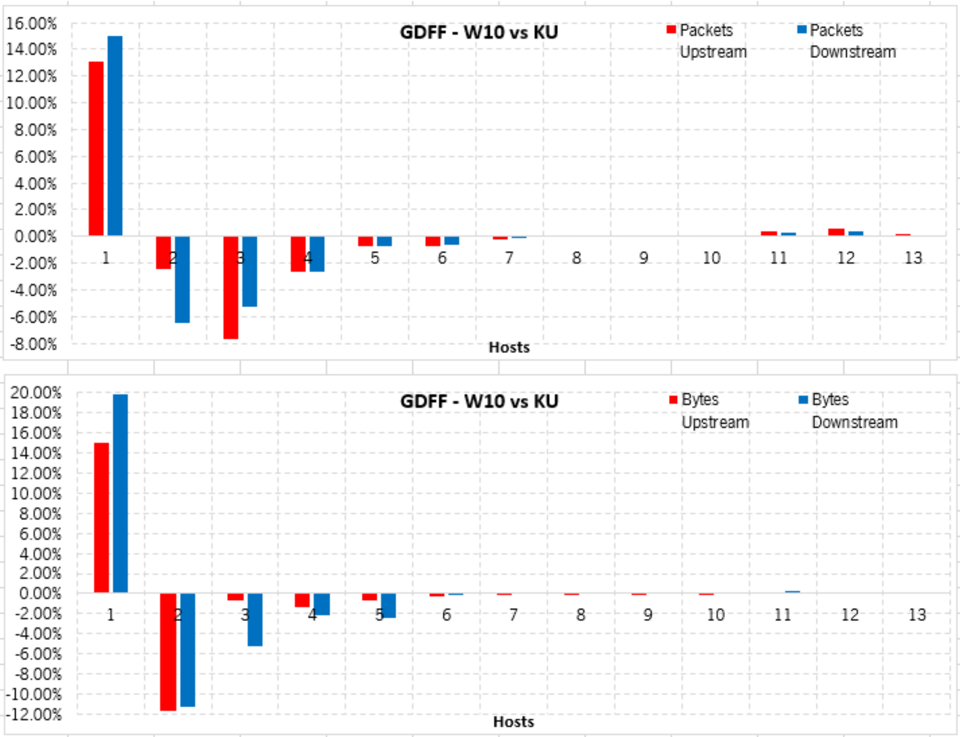


Figure 23: NC2W10GDFF versus NC2KUGDFF (normalized values) (Packets – top; Bytes – bottom)

Finally, the behavior observed in Figure 23 for both packet and byte count is almost perfectly inverse of the scenario shown in Figure 21.

Based on the three examined figures, it is clear therefore that Hypothesis D does not hold. Given the proprietary character of SaaS implementation as well as encrypted communication exchanged between the SaaS host and the end-user client, it is only possible to speculate as far as the observed behavior is concerned. One of the primary suspects for such a behavior difference is related with the way the operating system handles communication, i.e., the IP stack implementation in the given operating system and interaction with load balancing DNS implementation [38]. Details of the IP stack used in W7 and W10 are not readily available for analysis, through being one of the core portions of the operating system, it is expected that the stack implemented in W10 is much newer than the one used in W7. Neither of the two Windows versions used in this study are optimized out-of-the-box (without user intervention into Windows registry and/or network card interface configuration parameters) for high speed (symmetric 1Gbps FTTx) or high latency (mobile- or satellite-based) Internet access. KU IP stack is better optimized to typical high-speed Internet access scenarios prevalent today, better adapting to high-latency connections and providing better default buffering schemes suitable for high-speed Internet connections. There are no published research papers for reference on TCP/IP stack implementation details, especially in proprietary operating systems (Microsoft Windows), though variety of configuration options and online forum discussions on performance optimizations provide some insight into black-box performance differences between different versions of operating systems. There are though papers examining the performance of TCP/IP stack implementations in individual operating systems. [39]

The same analyses performed for GC and IE (where supported) in NC2 exhibit similar differences between individual operating systems. The observed behavior variations between individual operating systems also hold true for WO for each examined browser under the same network configuration (NC2). Changes in network configuration, namely for NC3 do not change the observed failure of Hypothesis D.

4.5 Hypothesis E: SaaS behavior is browser independent

Similar to Hypothesis D, it is also expected that the SaaS behavior is also browser-independent for one and the same operating system, i.e., does not depend on the type, version, and implementation details of the given browser. The SaaS elements in use on the end-client side are expected not to be resource intensive, requiring vast amounts of local hardware resources to operate the given cloud-based application. After all, one of the promises of cloud-based SaaS applications is their ability to be executed under any environment as long as it can support a basic browser with Internet access.

Note that the following analysis was performed for SaaS GD under W10 in NC2, with three target browsers, i.e., FF, GC, and IE. From all operating systems W10 was chosen due to the fact that is the only operating system that support all three browsers and Windows latest version.

Individual traces were collected and normalized using the same methodology described in section 4.4, with the resulting packet and byte count differences shown in the following figures:

- Figure 24 for comparison of SaaS GD, under W10 in NC2, under FF and GC,
- Figure 25 for comparison of SaaS GD, under W10 in NC2, under FF and IE,
- Figure 26 for comparison of SaaS GD, under W10 in NC2, under GC and IE.

Looking at the top portion of Figure 24, it is clearly visible that the behavior of FF and GC under the same operating system (W10) and in the same network configuration (NC2) is substantially different. The first SaaS host (host number 1) generates many more packets (~8% more) under GC when compared with FF. Higher order SaaS hosts (number 2, 3, and 4) generate more packets under FF. Similar observation can be made for the byte count, as shown in the bottom portion of Figure 24.

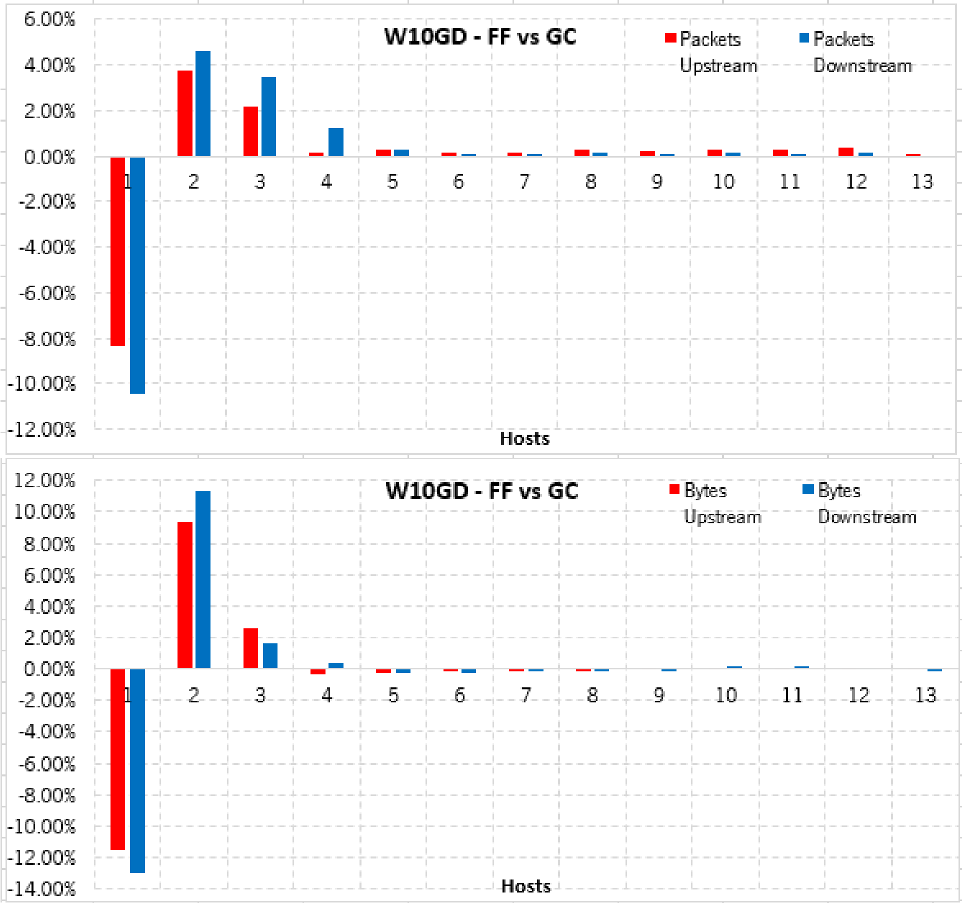


Figure 24: NC2W10GDFF versus NC2W10GDGC (normalized values) (Packets – top; Bytes – bottom)

Figure 25 exhibits similar behavior for SaaS GD when compared with Figure 24, in that (a) the first SaaS host under IE generates more data (packet and byte count wise) than under FF, though the data generated in the upstream (packet and byte count wise) values are smaller (~2 and ~4%, respectively), (b) SaaS hosts number 2 and 3 also generate more packets and bytes under FF, and (c) any differences for all the other hosts can be safely disregarded, given the differences well below the 1% threshold.

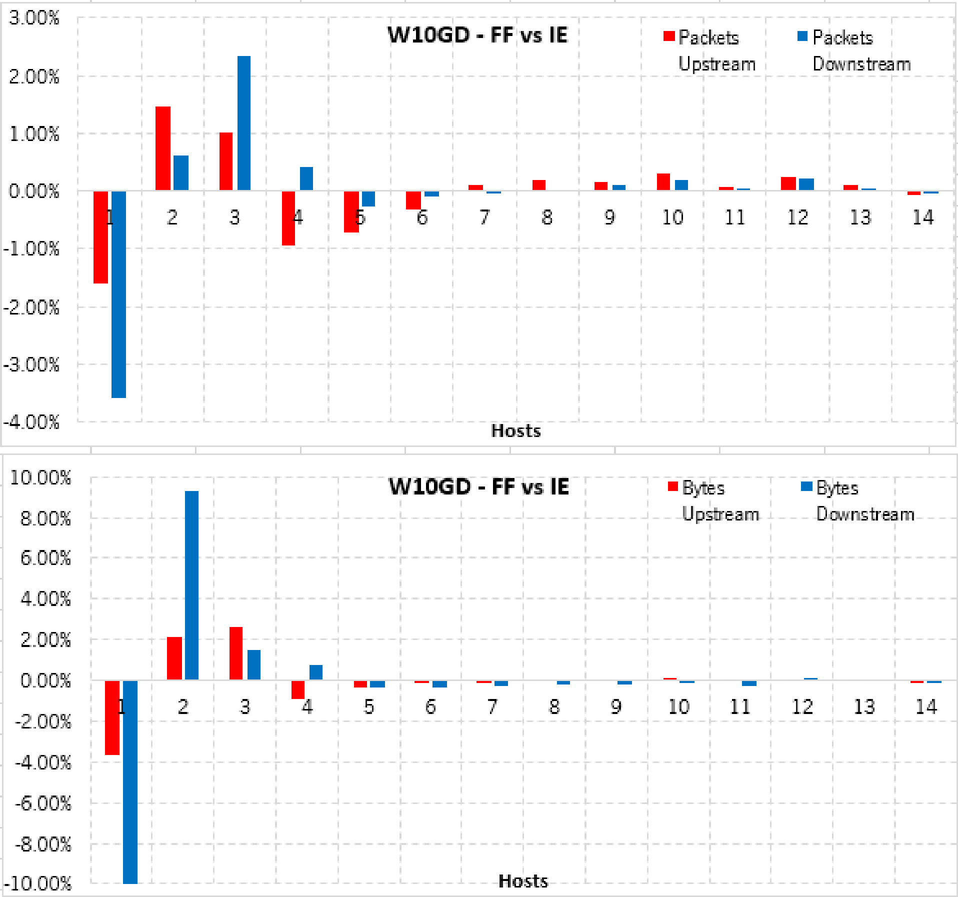


Figure 25: NC2W10GDFF versus NC2W10GDIE (normalized values) (Packets – top; Bytes – bottom)

Finally, given the behavior observed for GC versus FF, as well as FF versus IE, it can be expected that when comparing GC versus IE, the primary SaaS host will generate more packets and bytes under GC, with the higher order SaaS hosts (number 2 onwards) generating slightly more data under IE. This anticipated behavior is shown in Figure 26.

Therefore, based on the results presented in Figure 24, Figure 25, and Figure 26, Hypothesis E does not hold under W10 in NC2, with individual browsers exhibiting different behavior in their interaction with the SaaS hosts. In the case of GC these behavior differences may be explained with the use of QUIC transport protocol (UDP based) versus TCP-based transport protocol used by FF and IE. However, the two browsers using TCP-based transport protocol exhibit behavior differences, though their amplitude is much smaller when comparing either one of them against GC. The top portion of Figure 25 shows packet count differences on the order of ~3% or below for the primary SaaS host, while the byte count differences vary between ~4% and bellow for upstream and ~10% and bellow for downstream directions, respectively.

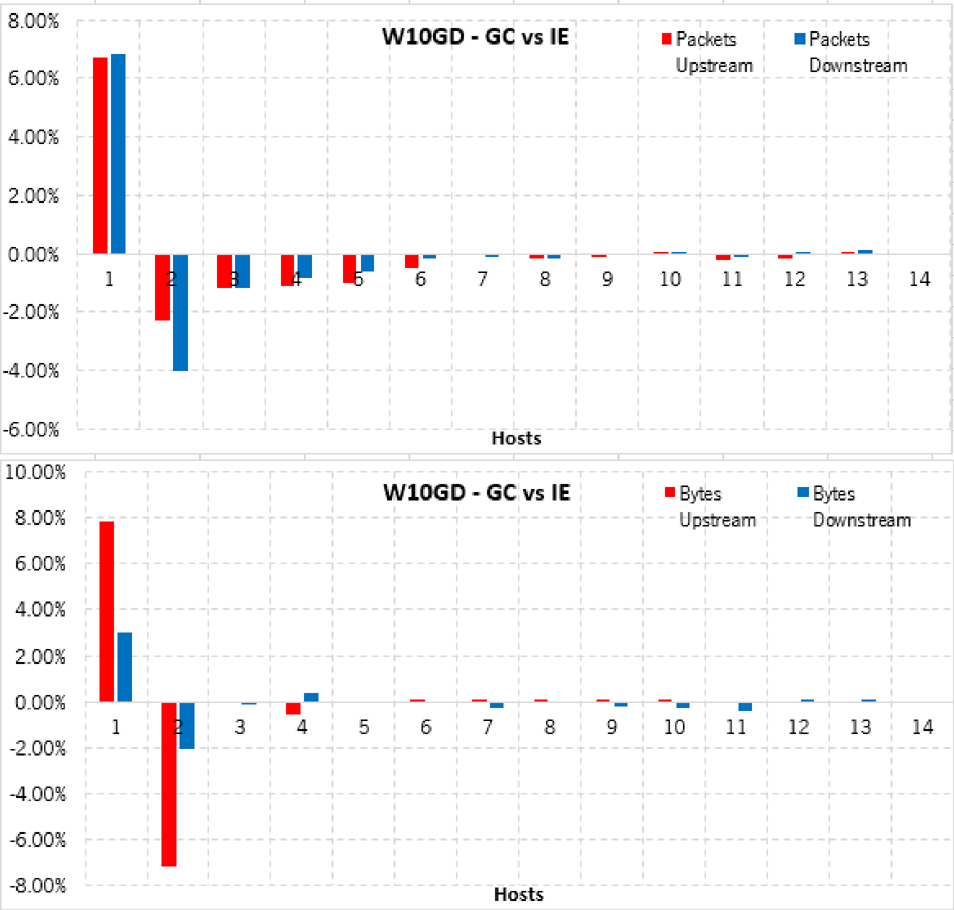


Figure 26: NC2W10GDGC versus NC2W10GDIE (normalized values) (Packets – top; Bytes – bottom)

The same analyses performed for W7 and KU in NC2 exhibit similar differences between individual browsers. The observed behavior variations between individual browsers also hold true for WO for each examined operating system under the same network configuration (NC2). Changes in network configuration (NC3) do not change the observed failure of Hypothesis E – the detailed results are not included due to the large volume of data.

4.6 Hypothesis F: SaaS behavior is network independent

Similar to Hypotheses D and E, it is expected that the SaaS behavior is network-independent for one and the same operating system and browser, i.e., does not depend on the network configuration. Two different network configurations were examined in this study, i.e., NC2 and NC3, as outlined in section 3.2, with the common operating system (Windows 7) and browsers (GC, FF, and IE). To assess the validity of this hypothesis, Windows 7 with FF running GD was selected, primarily to avoid the use of QUIC protocol. This hypothesis is especially crucial given the promise of SaaS to be a cross-platform, cross-browser, as well as cross-network solution, available to end users with low and high speed wired access as well as various forms of wireless access (primarily, 3G/4G-based).

The analysis methodology similar to the one used in sections 4.4 and 4.5 was applied in this case, as well, resulting in the packet and byte differences shown in Figure 27. The packet count in NC2 scenario is approximately 8% for the first host (upstream and downstream) when compared with NC3 scenario. Hosts 4 and 5 in NC3 scenario exchange ~2% more packets than the respective hosts in NC2 scenario, while the remaining hosts can be disregarded, with the packet count exchange volume difference of less than 1% (predefined threshold).

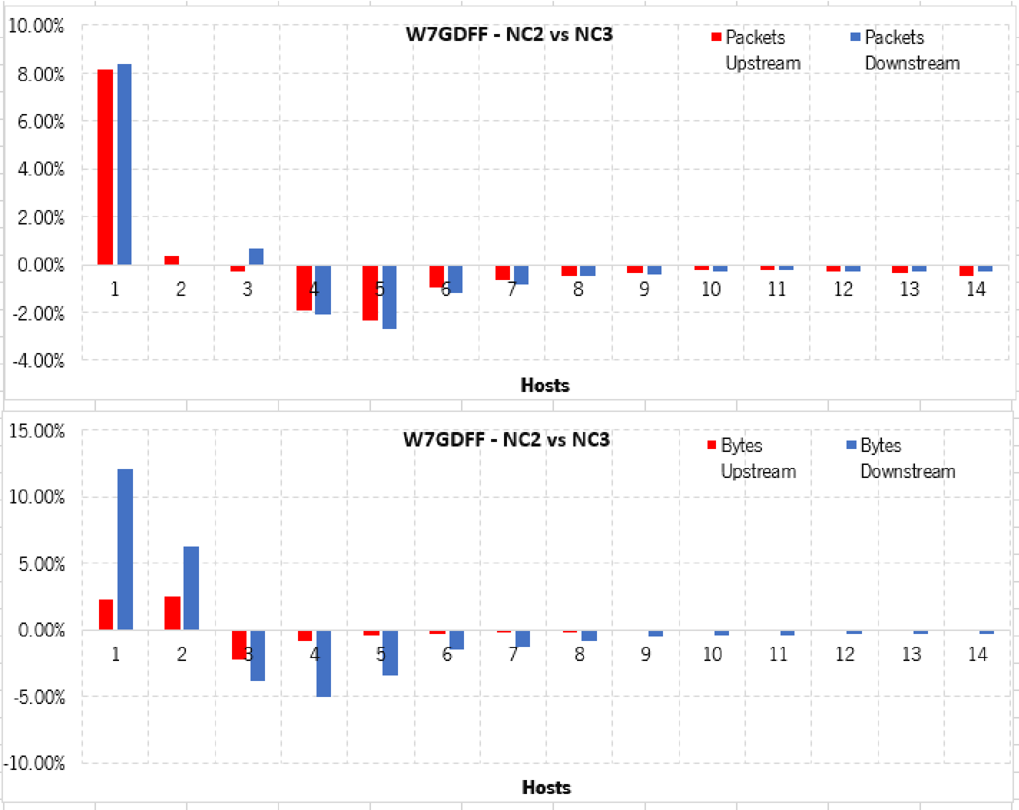
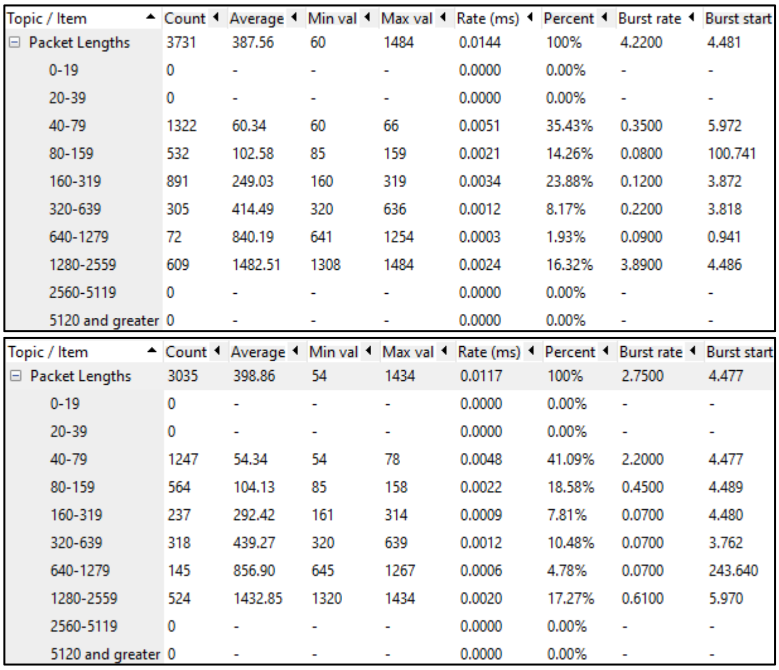


Figure 27: NC2W7GDFF versus NC3W7GDFF (normalized values) (Packets – top; Bytes – bottom)

The byte count is much more interesting. The upstream byte counts in NC2 are approximately 2% higher for hosts 1 and 2, while the upstream byte count is 2% higher for host 3 in NC3. The remaining upstream byte counts are comparable for these two scenarios. The downstream byte counts are a completely different story, though. Host number 1 in NC2 generated 12% bytes and host number 2 around 6% more bytes than the respective hosts in NC3. On the other and, host number 3 generated ~4% more, host number 4 generates ~5% more, and host number 5% generates around 3.5% more bytes in NC3 scenario when compared with respective hosts in NC2 scenario, resulting in a completely different characteristics of downstream traffic distribution between NC2 and NC3, with the upstream traffic distribution matching both scenarios more closely.

Taking the packet and byte difference profiles, it is obvious to conclude that packet size distributions in both networks must be different to produce such a large packet count difference while exhibiting such a small byte count difference in the upstream. This leads to conclusion that the NC3 features larger average packet size (fewer packets transmitted but resulting byte count delta is smaller) in the upstream when compared with NC2 scenario. Similarly, it can be concluded that the average packet size for the downstream direction for NC2 scenario is larger than in NC3 scenario, given that ~8% difference in packet count translates in ~12% in byte count difference between scenarios.

These observations are further substantiated by results shown in Figure 28 and Figure 29 for the examined scenarios in NC2 and NC3 (from Wireshark[1]).

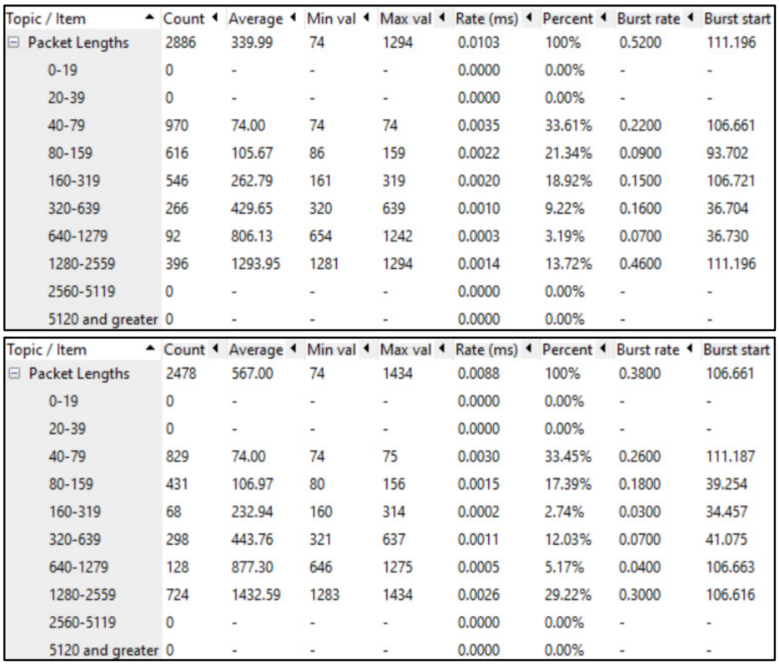


Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
⊟ Packet Lengths	3731	387.56	60	1484	0.0144	100%	4.2200	4.481
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	1322	60.34	60	66	0.0051	35.43%	0.3500	5.972
80-159	532	102.58	85	159	0.0021	14.26%	0.0800	100.741
160-319	891	249.03	160	319	0.0034	23.88%	0.1200	3.872
320-639	305	414.49	320	636	0.0012	8.17%	0.2200	3.818
640-1279	72	840.19	641	1254	0.0003	1.93%	0.0900	0.941
1280-2559	609	1482.51	1308	1484	0.0024	16.32%	3.8900	4.486
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
⊟ Packet Lengths	3035	398.86	54	1434	0.0117	100%	2.7500	4.477
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	1247	54.34	54	78	0.0048	41.09%	2.2000	4.477
80-159	564	104.13	85	158	0.0022	18.58%	0.4500	4.489
160-319	237	292.42	161	314	0.0009	7.81%	0.0700	4.480
320-639	318	439.27	320	639	0.0012	10.48%	0.0700	3.762
640-1279	145	856.90	645	1267	0.0006	4.78%	0.0700	243.640
1280-2559	524	1432.85	1320	1434	0.0020	17.27%	0.6100	5.970
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Figure 28: NC2W7GDFF scenario, downstream (top) and upstream (bottom) packet size statistics

[1] https://www.wireshark.org/docs/wsug_html/#ChStatistics



Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
⊟ Packet Lengths	2886	339.99	74	1294	0.0103	100%	0.5200	111.196
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	970	74.00	74	74	0.0035	33.61%	0.2200	106.661
80-159	616	105.67	86	159	0.0022	21.34%	0.0900	93.702
160-319	546	262.79	161	319	0.0020	18.92%	0.1500	106.721
320-639	266	429.65	320	639	0.0010	9.22%	0.1600	36.704
640-1279	92	806.13	654	1242	0.0003	3.19%	0.0700	36.730
1280-2559	396	1293.95	1281	1294	0.0014	13.72%	0.4600	111.196
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
⊟ Packet Lengths	2478	567.00	74	1434	0.0088	100%	0.3800	106.661
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	829	74.00	74	75	0.0030	33.45%	0.2600	111.187
80-159	431	106.97	80	156	0.0015	17.39%	0.1800	39.254
160-319	68	232.94	160	314	0.0002	2.74%	0.0300	34.457
320-639	298	443.76	321	637	0.0011	12.03%	0.0700	41.075
640-1279	128	877.30	646	1275	0.0005	5.17%	0.0400	106.663
1280-2559	724	1432.59	1283	1434	0.0026	29.22%	0.3000	106.616
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Figure 29: NC3W7GDFF scenario, downstream (top) and upstream (bottom) packet size statistics

In conclusion, Hypothesis G does not hold for NC2 and NC3, most likely due to different access network technology being used in these networks (FTTH versus mobile) as well as different L3 protocol in use (IPv4 in NC2 case, and IPv6 in NC3 case). There are various performance and operational differences between IPv4 and IPv6 protocols, leading to different packet size distributions as well as throughputs achievable. The underlying network access technology impacts also latency and achievable TCP performance in the case of examined SaaS (increased and unstable latency leads to TCP throughput degradation).

4.7 Hypothesis G: Preference for IPv6 traffic

Despite the availability of IPv6 for a number of years (IPv6 was developed back in 1998 to deal with the long-anticipated problem of IPv4 address exhaustion), the majority of residential networks still operate using NATing with IPv4 as the public address for the given home gateway. In such cases, the operating system has only a single way to reach the given SaaS host, i.e., using IPv4 transport. Network configuration scenarios NC1 and NC2 cover this option, where only a single public IPv4 address is available.

The network configuration scenario NC3 features the mobile (LTE) uplink, providing the end user with both IPv4 and IPv6 public addresses, allocated (in this case) from the pool of T-Mobile public addresses. In the case of IPv4 address, a NAT function is used, mapping local (non-routable address space [35]) into a single public IPv4 address, following the model used in most of the residential access networks. In the case of IPv6 address, no NAT function is needed, providing a truly public and routable IPv6 address to the end user device.

By default, most of the modern operating systems favor IPv6 global unicast addresses over IPv4 addresses, effectively showing the preference for any communication towards IPv6 protocols. The said protocol preference follows [40] and uses a prefix table to determine which address to use when multiple addresses are available for a Domain Name System (DNS) name. For example, if a specific host name resolves to both IPv4 and IPv6 address, IPv6 is used for communication with the said specific host. Obviously, if the DNS entry for the given host provides only a single (IPv4 or IPv6) address, there is no selection to be made. With both IPv4 and IPv6 available in NC3, it is therefore expected that most (if not all) communication with the SaaS hosts will take place over IPv6.

To test this hypothesis, individual traces were collected under all examined operating systems and browsers for different SaaS in NC3 were examined for the presence of IPv4 and IPv6 traffic and data exchange (number of packets and bytes) was then compared. Traces collected for NC2 and NC1 were omitted, given the IPv4-only transport options available in these network configuration scenarios.

Figure 30 and Figure 31 show the packet and byte count for NC3KU scenarios, respectively, with different browsers and SaaS hosts. There are a few observations that can be made.

- For WO, both examined browsers feature both IPv4 and IPv6 traffic, with IPv6 packet count roughly 2.5 times the number of IPv4 packets exchanged. The traffic volume follows similar ratio for IPv6 and IPv4 traffic for this particular SaaS.
- For GD, the situation is more interesting, with GC featuring almost exclusively IPv6 traffic only. FF uses both IPv4 and IPv6 traffic for this SaaS, though the IPv6 to IPv4 traffic ratio is much higher than for WO, reaching 5.2 for packet count and 3.8 for byte count.
- The WO under FF seems to be most packet and byte count intensive, while GD under GC is most efficient, featuring the lowest packet and byte count from all examined scenarios.

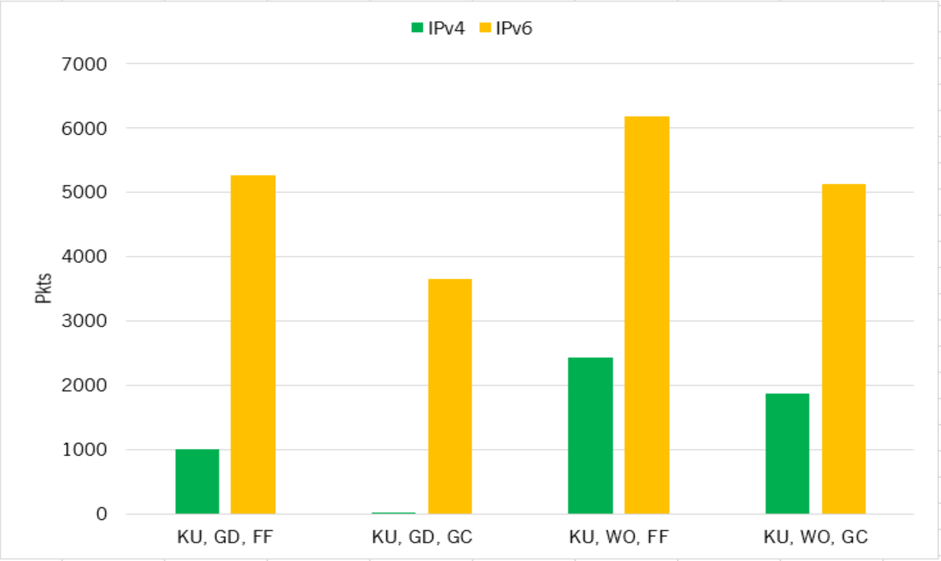


Figure 30: Packet count for NC3 scenarios for KU

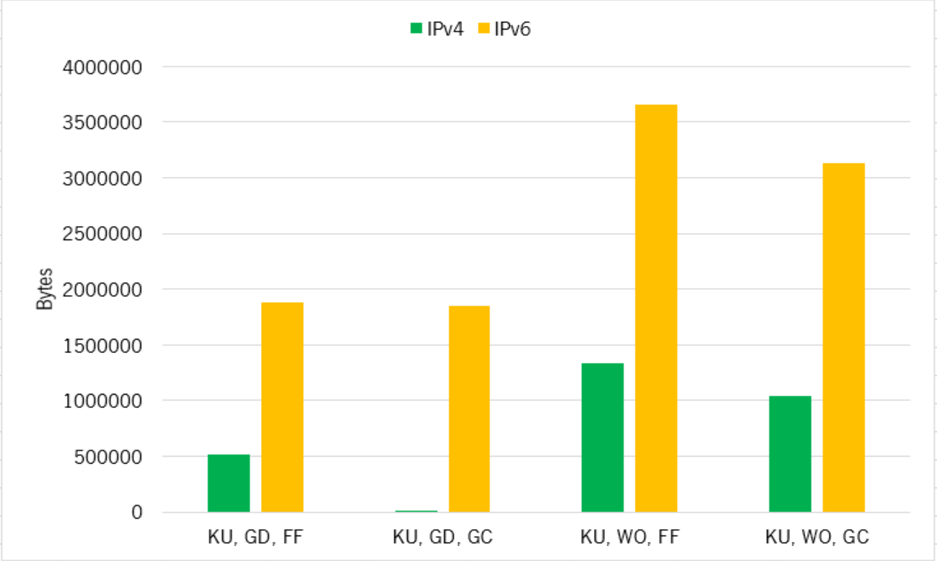


Figure 31: Byte count for NC3 scenarios for KU

Figure 32 and Figure 33 show the packet and byte count for NC3W10 scenarios, respectively, with different browsers and SaaS hosts. There are a few observations that can be made here as well, namely:

- Similar to NC3KU scenarios, for WO, both examined browsers feature both IPv4 and IPv6 traffic, though this time around the packet and byte count for WO under FF is comparable, while the IPv6 packet and byte count for GC for the same SaaS is roughly 2.5 times the number of IPv4 packets and byte exchanged.
- For GD under GC features almost exclusively IPv6 traffic only, similarly to what happened under KU. FF uses both IPv4 and IPv6 traffic for this SaaS, though this time around the IPv4 to IPv6 traffic ratio is much higher than for WO, reaching 5.2 for packet count and 3.8 for byte count, inverting the situation observed for the same browser and SaaS combination under KU.

- The WO under GC seems to be most packet and byte count intensive for W10, while GD under GC remains the most efficient, featuring the lowest packet and byte count from all examined scenarios.

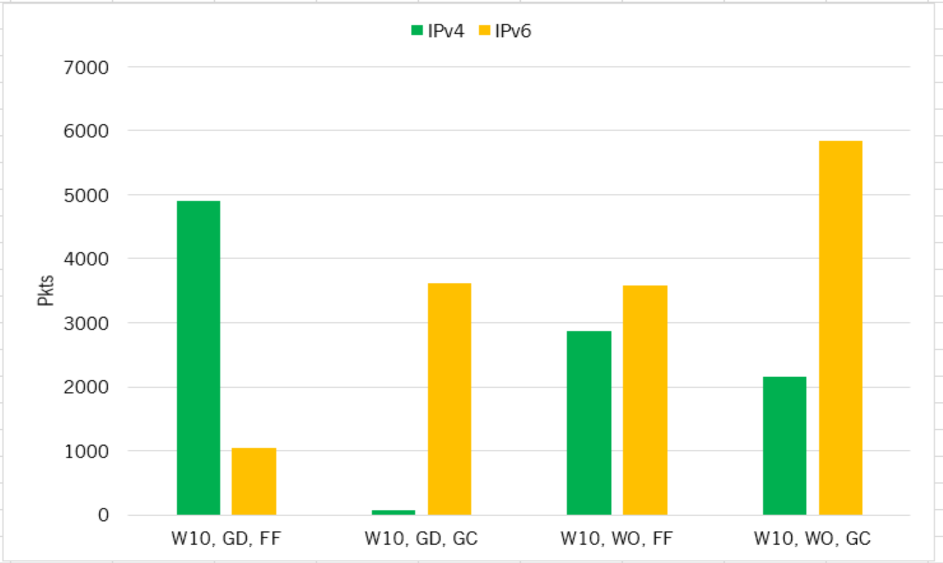


Figure 32: Packet count for NC3 scenarios for W10

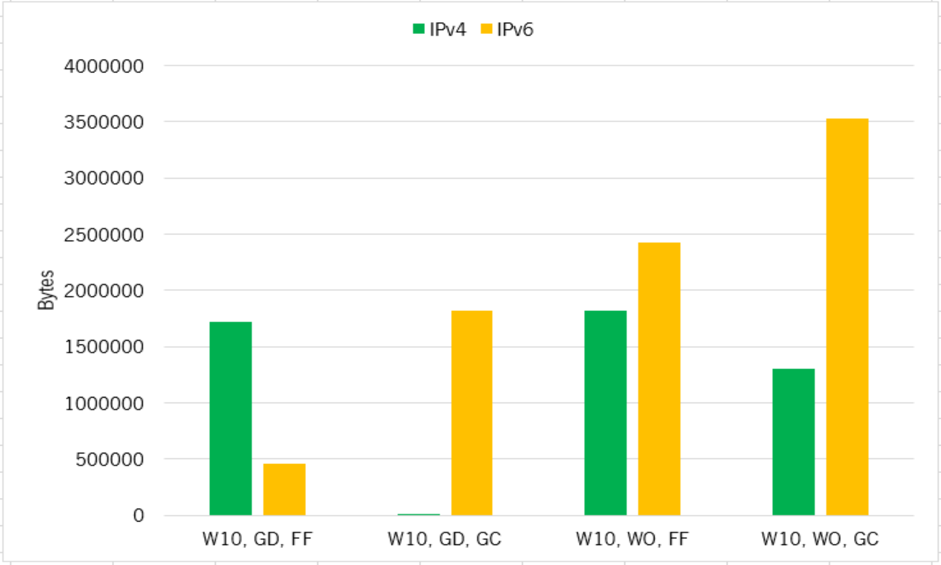


Figure 33: Byte count for NC3 scenarios for W10

In conclusion, Hypothesis G does not hold outside of GD under GC when running under KU or W10, where the volume of observed IPv4 traffic is small. The remaining browser, operating system, and SaaS combinations do feature substantial volume of IPv4 traffic, in some cases surpassing the volume of the observed IPv6 traffic. Given the preference for IPv6 at the operating system level, it is not clear at this time why there is such a large share of IPv4 traffic still observed. There are a few possibilities here.

- DNS records for specific host addresses still heavily skewed towards IPv4 addressing scheme, where the load balancing operation of DNS records returns more IPv4 than IPv6 address hits.

- Smaller number of SaaS hosts with public IPv6 addresses when compared to the same SaaS hosts with IPv4 addresses, forcing more traffic effectively towards IPv4 [41].
- Period inaccessibility of IPv6 addressed SaaS hosts, where the network path may become interrupted for unexplained reasons, causing the client side and the SaaS host side to fall back to IPv4 transport mechanism.

It is worth nothing, though, that the share of IPv6 traffic in the collected traces is substantial, providing a glimpse into the future of IP-based networking, where the total share of all IPv6 traffic increases steadily as more and more services become dual-stacked (have both IPv4 and IPv6 addresses assigned) or IPv6 only, reachable from legacy IPv4 clients only through some of the IP tunneling mechanisms [42][43].

4.8 Hypothesis H: QoE is network-independent for SaaS

The Quality of Experience (QoE) is typically defined as a measure of the satisfaction (delight or annoyance) of a customer's experiences with any kind of service, including the SaaS applications in this case. QoE focuses on the entire service experience, and as such, it is considered a holistic metric for assessing end user experience with the given platform, solution, system, appliance, or application.

The concept of QoE is very broad and unfortunately – not defined in a very concise manner. ITU-T P.10 Recommendation [44] and Qualinet [45] formalized the QoE definition as "*The degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and / or enjoyment of the application or service in the light of the user's personality and current state.*", and it has enjoyed a decent level of adoption in the communication industry to date.

In simple terms, QoE is a purely subjective measure from the user's opinion of the overall quality of the service provided, by capturing and focusing mostly on people's "aesthetic" perception in terms of (for example) image and sound quality, application responsiveness, reliability, etc. As such, there is no fixed definition of aspects of end user experience that should be measured under QoE, and most of them are very subjective (e.g., sound quality, responsiveness, attractiveness), requiring studies to be conducted on a test group to properly assess the given aspect of QoE.

QoE most often attempts to build on QoS metrics as well, providing means of assessing at least some of the QoE aspects in a more objective manner. Typically, packet loss, latency, jitter and average throughput are used to assess service availability and responsiveness, directly translating into the end user perception. These aspects become especially critical in the cases of less reliable, bandwidth constrained, or ad-hoc networks. An increasing number of Internet users, especially in the residential setting, uses various forms of Internet access mobility services, relying on mobile phone carrier (3G/LTE/5G) or even satellite links for Internet connectivity. In such a setting, SaaS may have to contend with less reliable Internet connectivity, putting substantially more focus on reliable synchronization between the end client application and the SaaS host(s). Furthermore, especially in the case of satellite-based Internet connectivity, the increased latency may have detrimental impact on SaaS operation and QoE as perceived by the end user.

4.8.1 Traffic Profile

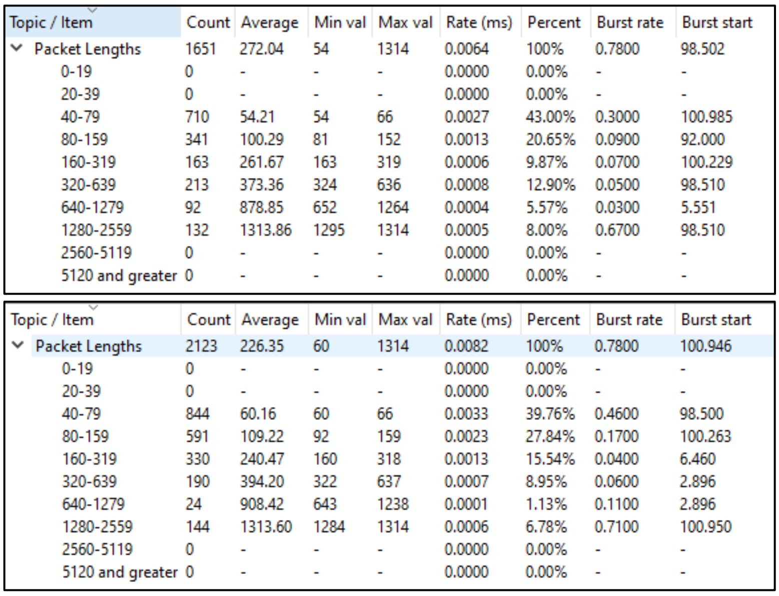
The mobile and satellite-based Internet access typically features a reasonably sized downstream capacity, providing enough bandwidth to use most common download-intensive applications (video streaming, video conferencing, etc.). However, it is the upstream capacity that is critical for SaaS operation, where limited bandwidth is typically provided due to the large aggregation (number of connected customers) and limited available spectrum. In short, these types of networks are highly asymmetric in terms of bandwidth, which may have impact on SaaS operation.

From the analysis of previous hypotheses (see section 4.3, collecting upstream and downstream traffic statistics), it is clear that the large share of all examined SaaS application, operating system, browser, and network scenario configurations feature substantially larger volume of upstream than downstream traffic. Effectively, when using specific SaaS, more data is uploaded than is downloaded, with just a handful of exceptions. This means, effectively, that SaaS behavior in upstream-capacity-constrained network scenarios is expected to be degraded when compared to fixed access networks, irrespective of whether they are fiber (FTTH) or copper (FTTC or xDSL) access based. To better understand the distribution of packets in upstream and downstream scenarios, the following analysis was performed for GD and SaaS WO applications in NC1 scenarios for W7 operating system.

For scenario NC1W7GDFF (used as an example for the following description), the majority of transmitted and received packets (see Figure 34, top and bottom, respectively) are below 319 bytes in size (73.52% in upstream direction and 83.14% in downstream direction). Such a small packet size is likely a direct result of the way the application operates, sending frequently updates from the client to the SaaS host. With few changes to report most of the time, the resulting packet size is most often small, as observed in the collected packet traces. Large packet sizes are only likely to occur when the client needs to download a lot of content in a short period of time (for example, a newly inserted image, a new table, etc.), or the client needs to update the SaaS host with a lot of locally created content (e.g., plenty of text pasted by end user, upload a locally inserted image or table, etc.).

Moreover, the process of frequent content saving noticed during tests seems to be suitable for unreliable network access, when accessing SaaS services via a mobile or satellite connection. In this case, the application needs to save local changes as often as possible, to avoid data loss.

Similar conclusions may be drawn for other scenarios (SaaS application, operating system, browser, etc., combinations) following a similar analysis process. The behavior of IE is typically substantially different compared to GC and FF. While in the IE, the SaaS client likely needs to download/update the complete SaaS GUI to the end user client which would explain the heavy mode in the downstream direction for large packets, in the GC and FF, the client seems to render locally resulting in sending frequent but smaller updates to the SaaS host. This also means that IE generates more downstream than upstream traffic and, therefore, rendering it less suitable for the upstream-constrained access networks (mobile and satellite-based ones).

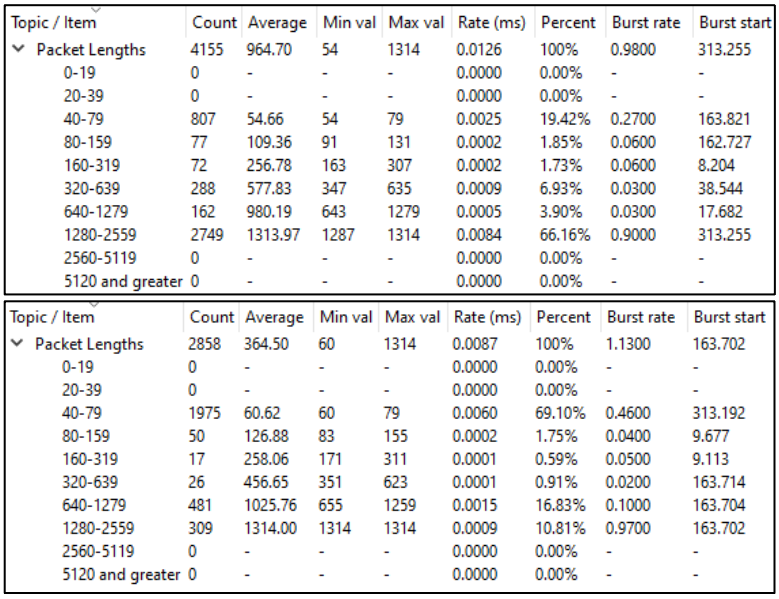


Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Packet Lengths	1651	272.04	54	1314	0.0064	100%	0.7800	98.502
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	710	54.21	54	66	0.0027	43.00%	0.3000	100.985
80-159	341	100.29	81	152	0.0013	20.65%	0.0900	92.000
160-319	163	261.67	163	319	0.0006	9.87%	0.0700	100.229
320-639	213	373.36	324	636	0.0008	12.90%	0.0500	98.510
640-1279	92	878.85	652	1264	0.0004	5.57%	0.0300	5.551
1280-2559	132	1313.86	1295	1314	0.0005	8.00%	0.6700	98.510
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Packet Lengths	2123	226.35	60	1314	0.0082	100%	0.7800	100.946
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	844	60.16	60	66	0.0033	39.76%	0.4600	98.500
80-159	591	109.22	92	159	0.0023	27.84%	0.1700	100.263
160-319	330	240.47	160	318	0.0013	15.54%	0.0400	6.460
320-639	190	394.20	322	637	0.0007	8.95%	0.0600	2.896
640-1279	24	908.42	643	1238	0.0001	1.13%	0.1100	2.896
1280-2559	144	1313.60	1284	1314	0.0006	6.78%	0.7100	100.950
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Figure 34: NC1W7GDFF packet lengths statistics, upstream and downstream (from top to bottom)

For scenario NC1W7WOFF, the majority of transmitted packets (see Figure 35, top) are above 319 bytes in size (76.99%), while the majority of received packets (see Figure 35, top) are below 319 bytes in size (71.44%). This represents a substantial change from the behavior observed in section 4.1.1, likely related with the heavier information upload towards the SaaS host from the end user client than in the case of SaaS GD. Similar conclusions apply to others scenarios.

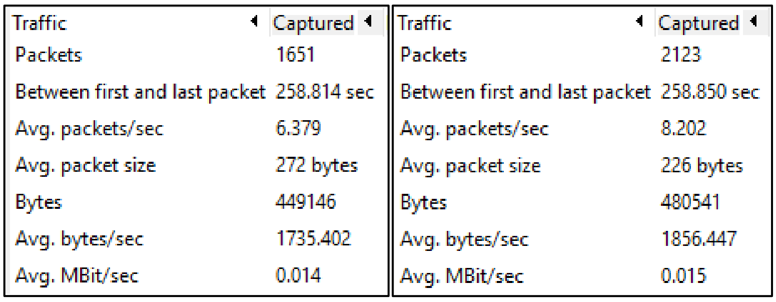


Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Packet Lengths	4155	964.70	54	1314	0.0126	100%	0.9800	313.255
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	807	54.66	54	79	0.0025	19.42%	0.2700	163.821
80-159	77	109.36	91	131	0.0002	1.85%	0.0600	162.727
160-319	72	256.78	163	307	0.0002	1.73%	0.0600	8.204
320-639	288	577.83	347	635	0.0009	6.93%	0.0300	38.544
640-1279	162	980.19	643	1279	0.0005	3.90%	0.0300	17.682
1280-2559	2749	1313.97	1287	1314	0.0084	66.16%	0.9000	313.255
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Packet Lengths	2858	364.50	60	1314	0.0087	100%	1.1300	163.702
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	1975	60.62	60	79	0.0060	69.10%	0.4600	313.192
80-159	50	126.88	83	155	0.0002	1.75%	0.0400	9.677
160-319	17	258.06	171	311	0.0001	0.59%	0.0500	9.113
320-639	26	456.65	351	623	0.0001	0.91%	0.0200	163.714
640-1279	481	1025.76	655	1259	0.0015	16.83%	0.1000	163.704
1280-2559	309	1314.00	1314	1314	0.0009	10.81%	0.9700	163.702
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Figure 35: NC1W7WOFF packet lengths statistics, upstream and downstream (from top to bottom)

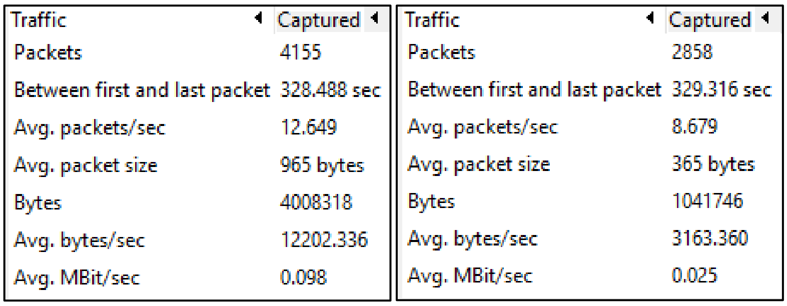
Traffic statistics for individual examined scenarios are shown in Figure 36 for NC1W7GDFF and in Figure 37 for NC1W7WOFF, respectively (from Wireshark[2]). Similar analysis was performed for other examined scenarios, with similar conclusions, though they were not included to keep the size of this document under control.



Traffic	Captured
Packets	1651
Between first and last packet	258.814 sec
Avg. packets/sec	6.379
Avg. packet size	272 bytes
Bytes	449146
Avg. bytes/sec	1735.402
Avg. MBit/sec	0.014

Traffic	Captured
Packets	2123
Between first and last packet	258.850 sec
Avg. packets/sec	8.202
Avg. packet size	226 bytes
Bytes	480541
Avg. bytes/sec	1856.447
Avg. MBit/sec	0.015

Figure 36: NC1W7GDFF traffic statistics for upstream and downstream directions (from left to right)



Traffic	Captured
Packets	4155
Between first and last packet	328.488 sec
Avg. packets/sec	12.649
Avg. packet size	965 bytes
Bytes	4008318
Avg. bytes/sec	12202.336
Avg. MBit/sec	0.098

Traffic	Captured
Packets	2858
Between first and last packet	329.316 sec
Avg. packets/sec	8.679
Avg. packet size	365 bytes
Bytes	1041746
Avg. bytes/sec	3163.360
Avg. MBit/sec	0.025

Figure 37: NC1W7WOFF traffic statistics for upstream and downstream directions (from left to right)

[2] https://www.wireshark.org/docs/wsug_html_chunked/ChStatSummary.html

4.8.2 Latency, Latency Variation (Jitter), and Packet Loss

The other aforementioned aspects of the mobile and satellite-based access networks are the connection latency (delay) and its variability. As an example, consider the ICMP statistics collected during execution of one of the NC3 scenarios, where a constant ping was executed to the `docs.google.com` SaaS host located at 2607:7700:0:1f:0:1:d073:881b address (see Figure 38).

```
Ping statistics for 2607:7700:0:1f:0:1:d073:881b:
    Packets: Sent = 1705, Received = 1657, Lost = 48 (2% loss),
Approximate round trip times in milli-seconds:
    Minimum = 46ms, Maximum = 1495ms, Average = 122ms
```

Figure 38: ICMP output for SaaS GD host (2607:7700:0:1f:0:1:d073:881b), NC3

Note the packet loss of ~2% being observed, which will obviously result in retransmission events on either side of the link. The latency (RTT) observed is also critical in this case, since it is varying wildly between 46 ms (best observed number) and 1495 ms (worst observed number) with the average of 122 ms.

The resulting ICMP latency was then plotted for all 1700 packets as it is shown in Figure 39 (blue). It can be observed that most of the time, the latency is below ~100 ms, though there are at least three major latency spikes, with measured latency exceeding 1000 ms and packet loss (plotted as -500 ms, for visualization purposes only).

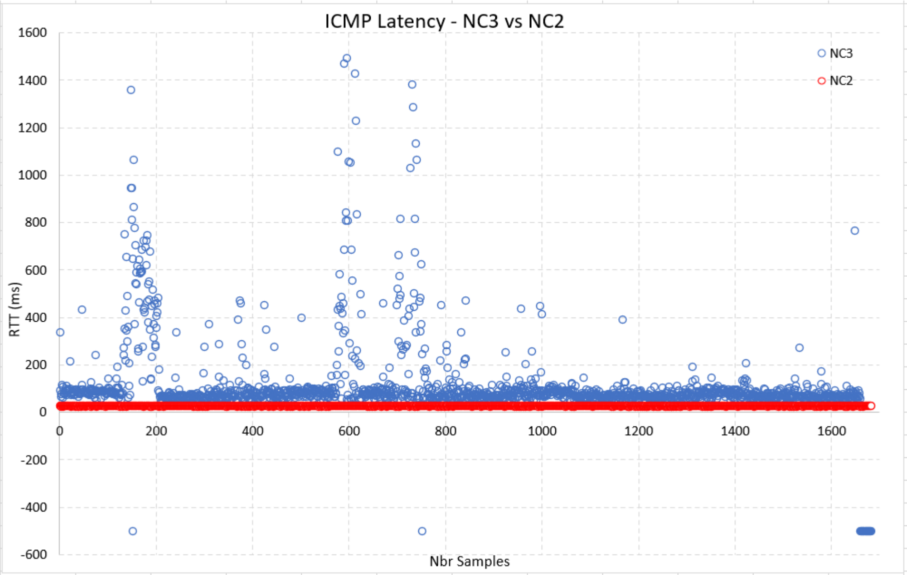


Figure 39: ICMP latency for SaaS GD in one of NC3 scenario (blue) and a fixed access network NC2 (red)

It is worth to note that, latency (one way delay) exceeding 100 ms /150 ms is typically considered QoE impacting for real time applications. Even though the examined SaaS applications are not included in this category, the observed latency values (RTT) are well below the threshold of 200 ms /300 ms most of the time. Moreover, when the latency threshold was exceeded, the QoE started to be degraded, leading to dissatisfaction with the SaaS application. Effectively, the end user can tolerate a certain level of latency in the application, though once exceeded, the resulting experience deteriorates rapidly.

For comparison purposes, the latency for a fixed access network (NC2) was measured, with the summary shown in Figure 40. The resulting 1700 packet latency plot is also shown in the previous Figure 39 (red). It can be observed that the measured latency is substantially lower, with the average of around 25 ms, and even the measured maximum latency below 27 ms, with the latency deviation of only 230 µs. This level of performance, including zero packet loss, provides a much more stable environment for SaaS application, resulting in substantially better QoE.

```
Ping statistics for 2607:7700:0:1f:0:1:d073:881b:
    Packets: Sent = 1704, Received = 1704, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 25.630ms, Maximum = 26.553ms, Average = 25.750ms
```

Figure 40: ICMP output for SaaS GD host (2607:7700:0:1f:0:1:d073:881b), NC2

During the collection of individual packet traces in NC3 scenarios, several QoE-affecting observations were made, as summarized below. Note that these observations are mostly subjective, based on interaction with the given SaaS application either manually or through pre-recorded macro replay.

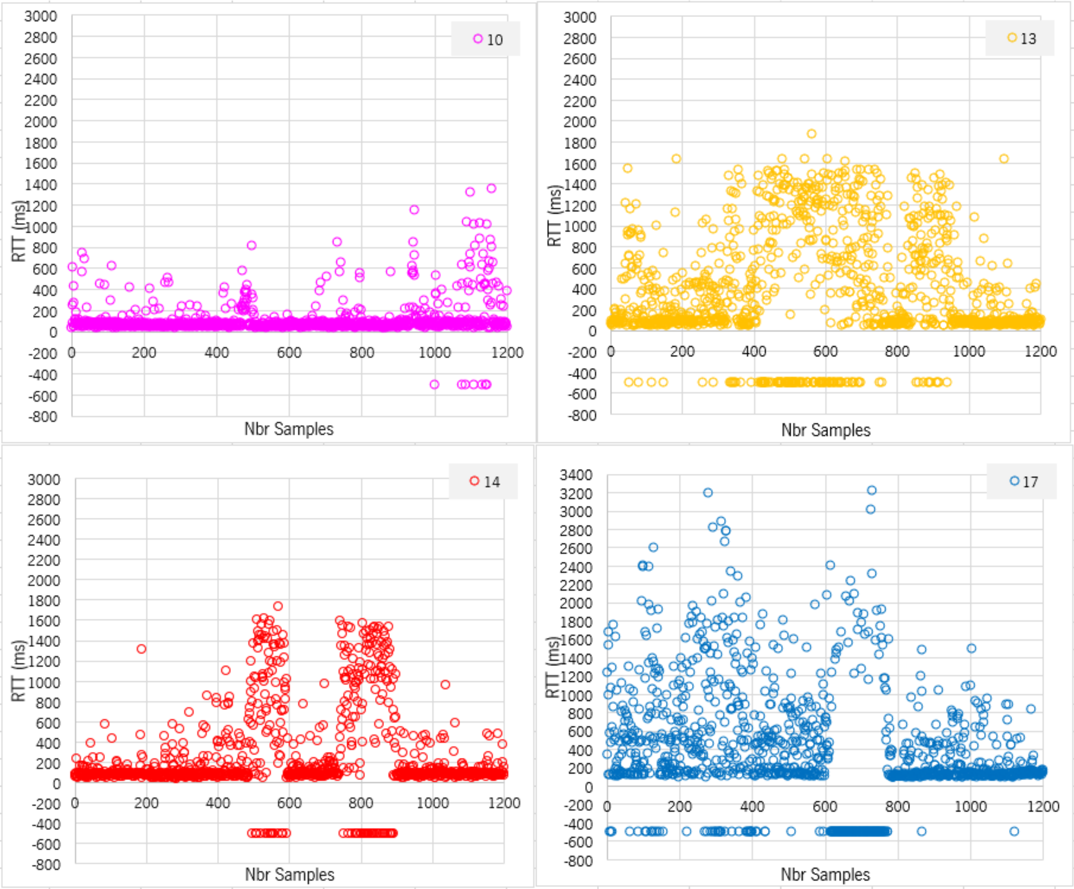


Figure 41: Latency and packet loss for SaaS GD, plotted for 10am, 1pm, and 5 pm (connectivity loss events shown as negative numbers)

There are several periods of the day and week, where the (likely) network congestion affects the ability to connect to and interact with any of the examined SaaS (GD or WO). During regular workdays, periods between 11 am and 1:30 pm resulted in very low SaaS responsiveness, where it would at times takes minutes for SaaS application to refresh the screen and catch up with, or even allow, the given set of actions, added text, formatting, etc. There is a positive correlation between the SaaS responsiveness, time of day, as well as the connectivity loss (plotted as negative numbers, -500), as shown in Figure 41.

It is immediately visible that along the day, there are completely different SaaS accessibility profiles, where at 10 am, we have reasonably stable latency and very few connection loss events, while the busy periods of the day (1 pm and 5 pm) show very high latency variability as well as extended period of connectivity loss, where the SaaS is simply not accessible to end user.

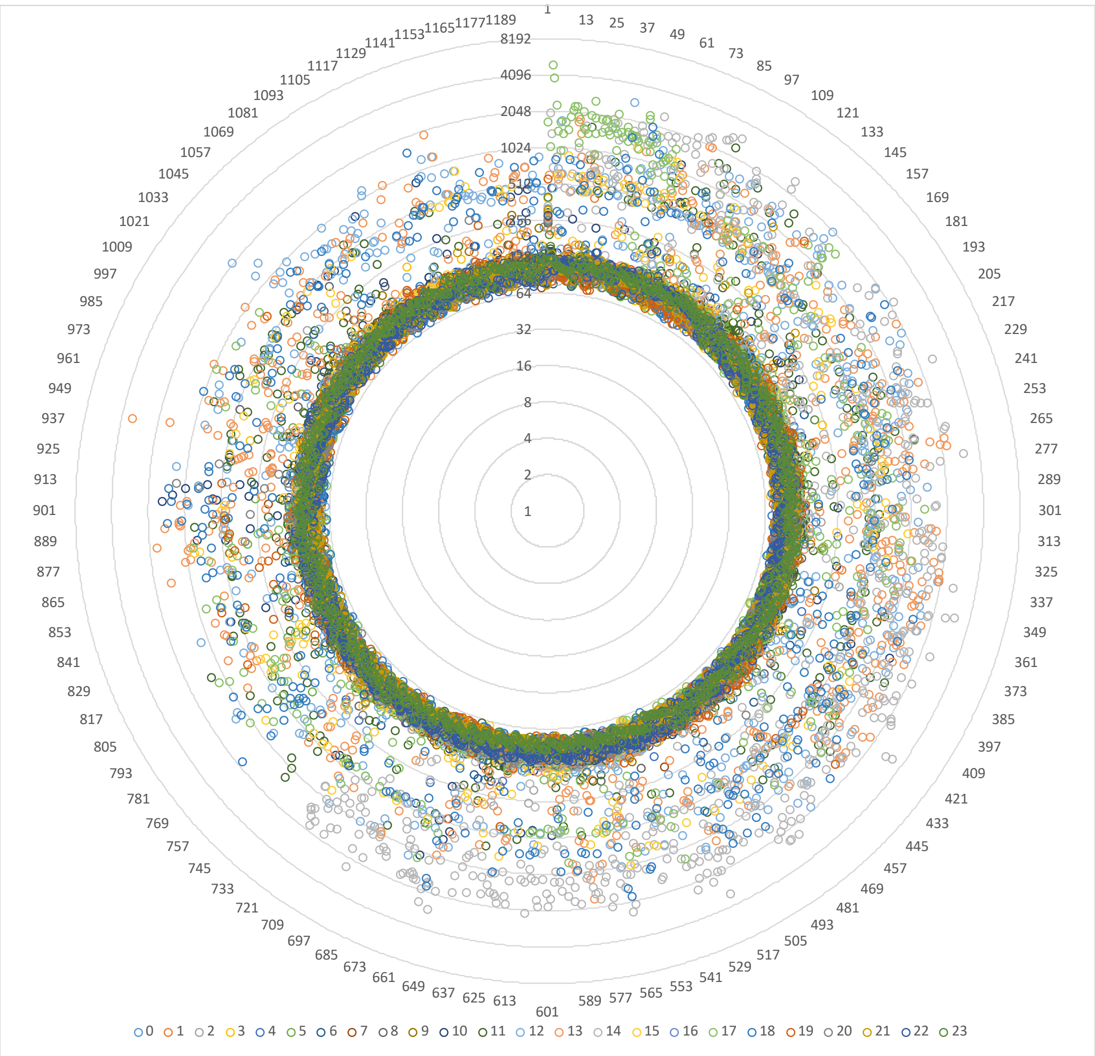


Figure 42: Network latency to SaaS GD host over period of 24 hours (weekday), NC3, IPv6

A plot of measured network latency to SaaS GD and SaaS WO hosts (expressed in units of ms) is shown in Figure 42 and Figure 43, respectively. All circular charts showing network latency use the following axes:

- Vertical axis represents measured network latency, expressed in units of ms. For improved readability this axis uses logarithmic scale (base 2).
- Circular axis (horizontal) represents sequential latency measurement sample number. The total of 1200 samples are represented in each chart.

Both SaaS hosts were IPv6 addressed. Note that maximum measured latency reached 5000 ms for SaaS GD and 4500 ms for SaaS WO. Clearly, such a large delay is detrimental to perceived QoE from the end-user perspective.

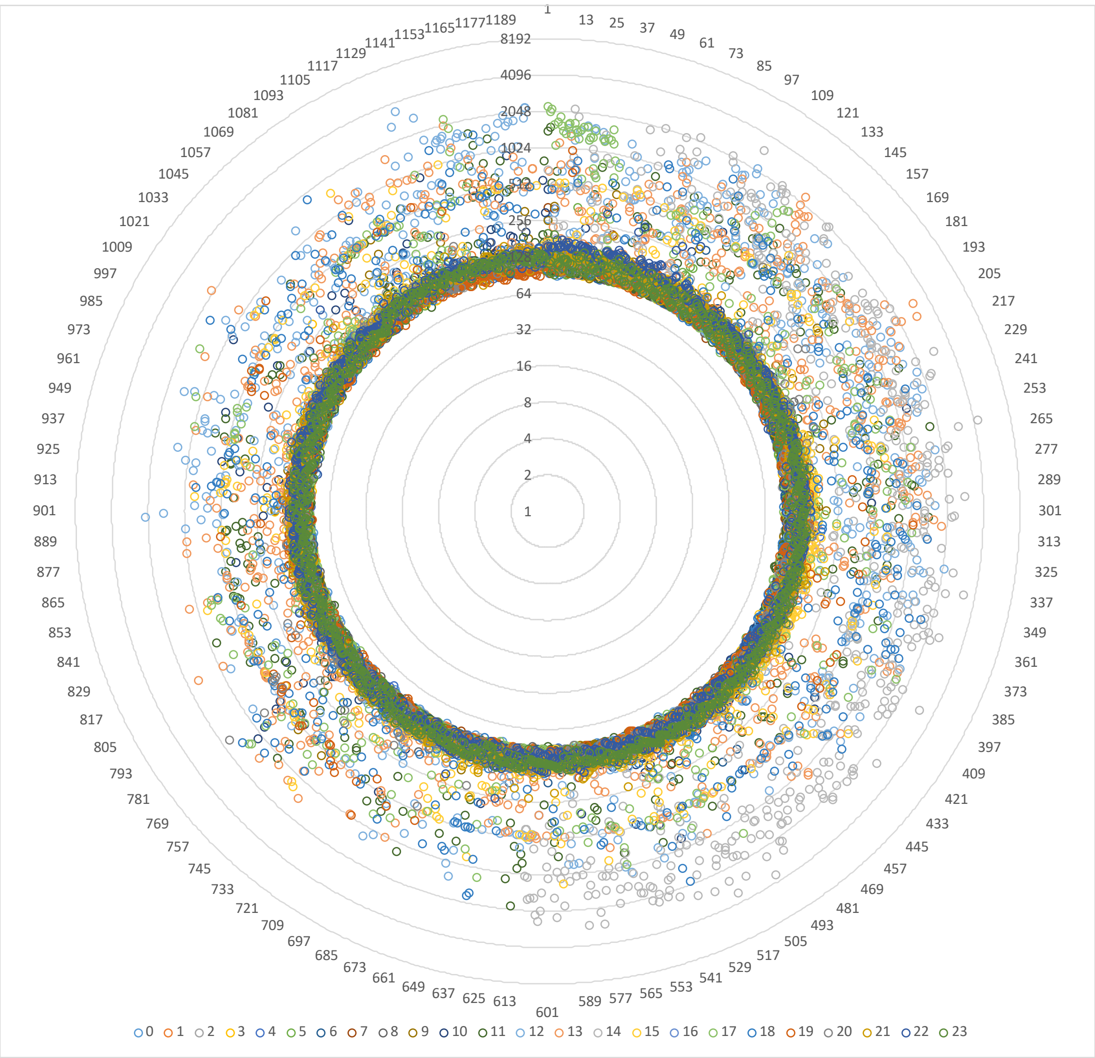


Figure 43: Network latency to SaaS WO host over period of 24 hours (weekday), NC3, IPv6

For comparison purposes, latency over the period of 24 hours was measured and plotted for the very same SaaS WO, i.e., GD and WO, but this time in NC2 configuration (fixed access network). The resulting latency is shown in Figure 44 for SaaS GD and Figure 45 for SaaS WO. What is immediately visible when comparing Figure 42 and Figure 44 for the very same SaaS but different network configurations, is that IPv4 latency is much lower (average latency of 64 ms for NC3 versus the average latency of 4 ms for NC2), along with much smaller maximum values (~6000 ms for NC3 versus ~780 ms for NC2). Similar conclusions can be drawn when comparing SaaS WO for NC3 (see Figure 43) and for NC2 (see Figure 45).

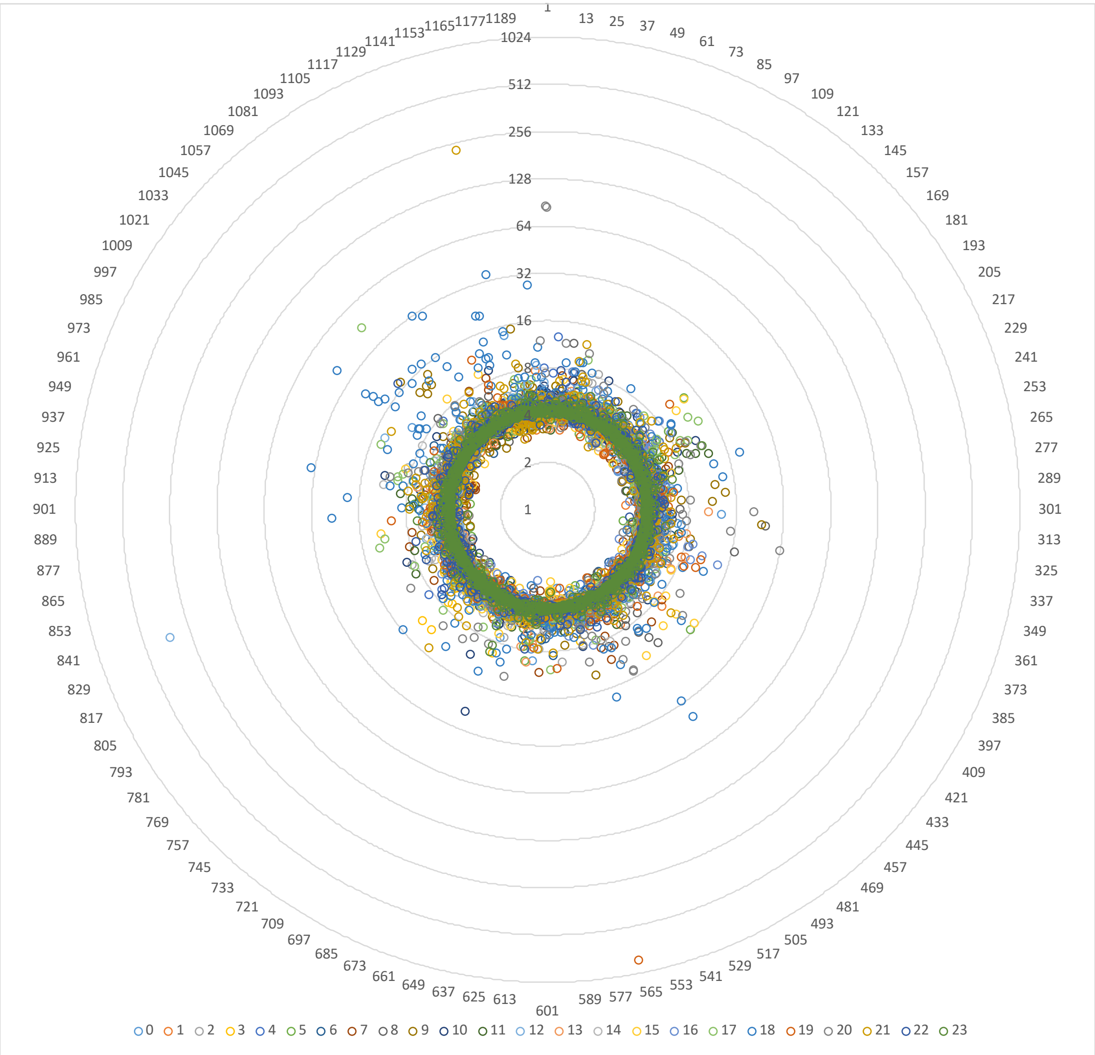


Figure 44: Network latency to SaaS GD host over period of 24 hours (weekday), NC2, IPv4

Furthermore, it was also observed that any attempts to access SaaS via mobile network on Friday evening resulted in connection failures most of the time, and times when connectivity was established, the latency rendered the SaaS application completely unusable. A more long-term study of the network latency over the period of weeks is needed to provide more conclusive observations of the impact of day of the week and time of time on SaaS operation.

During periods of network congestion, the (most likely) packet loss and resulting failed retransmissions lead to unexpected behavior of the SaaS application, causing failures to save reliability the edited document, data corruption, and most often – unexpected outcome of typical text editing operations.

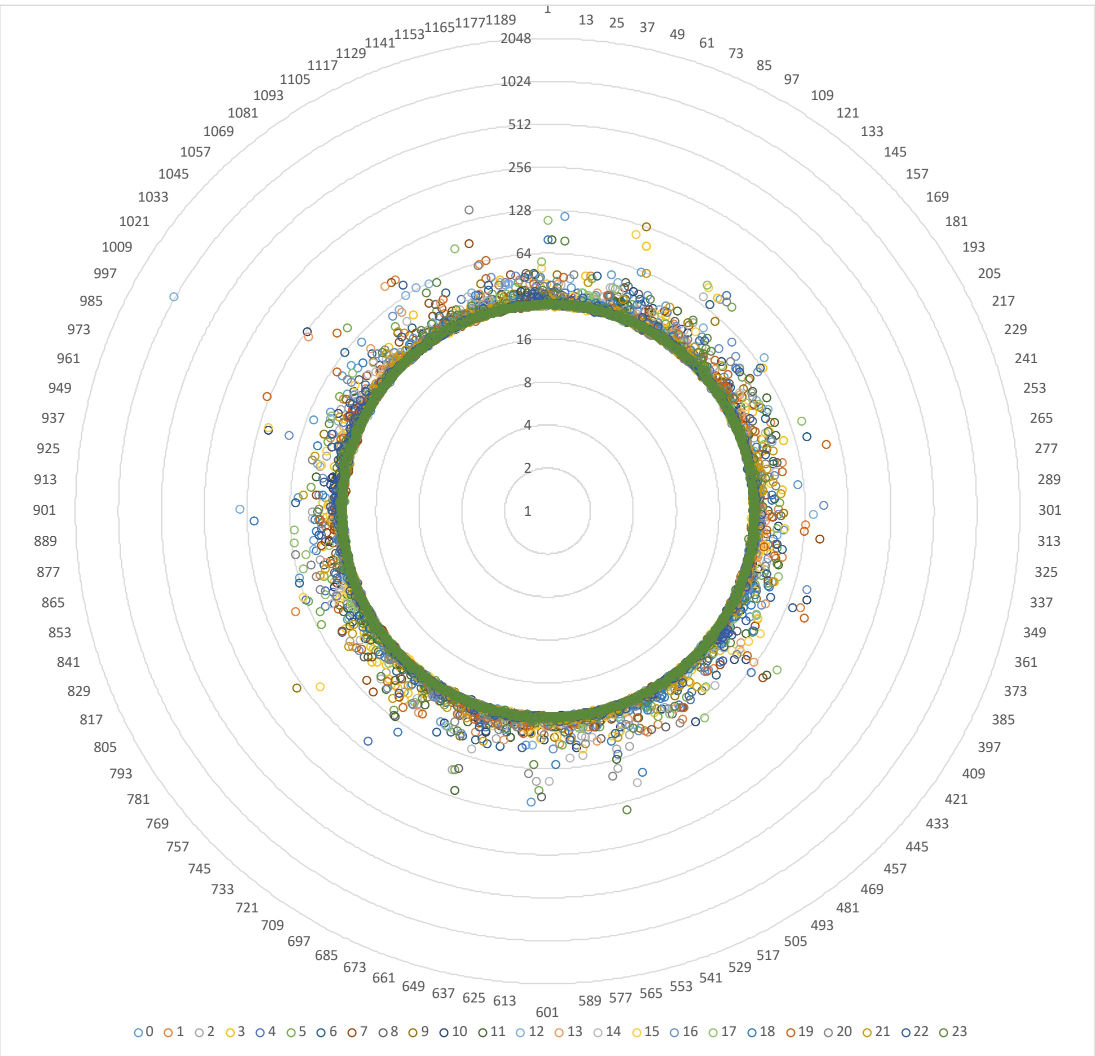


Figure 45: Network latency to SaaS WO host over period of 24 hours (weekday), NC2, IPv4

4.8.3 SaaS Failure Severity Levels

Three major levels of the perceived QoE associated with a SaaS application are hereby defined:

- **Severity level 1:** SaaS application remains usable, though there is potential for the end-user annoyance due to the diminished usability of the given application. At this level, there is perceived delay between end user introducing some data into the SaaS application, and the result displaying on the screen. Ideally, such a delay is non-existent, and the end user cannot distinguish a local application from a cloud-based application.
- **Severity level 2:** SaaS application remains usable though with observation period of access interruption and potential data loss. In this case, primarily due to the packet loss at the network transport layer, some of the data introduced by the end user may be lost (never makes it to the SaaS host(s)), resulting in the need to repeat actions (e.g., text formatting, adding text, image,

etc.). Depending on the number of such events, the end user may experience loss of confidence in the SaaS application and stability of the given system.

- **Severity level 3:** SaaS application is unusable most of the time. In this case, due to the packet loss and/or extreme latency (delay) at the network transport layer, the SaaS application remains mostly inaccessible (application times out, display access errors, etc.) and even when it is accessible – the usability factor is severely limited, with constant need to repeat actions and/or upload data, putting the operation and stability of the SaaS application in question.

When collecting traces in NC3, individual SaaS application operation artifacts were tallied and graded using the aforementioned severity grading system, with the resulting data presented in Table 35 and Table 36 for SaaS GD and SaaS WO, respectively.

Table 35: Artifacts observed in NC3 for SaaS GD

Severity Level	Nbr times observed #fails/#total			Total % of fails for SaaS GD	
1 **Annoyance**	FF	W7	4/11	28.6%	20/73 **27.4%**
		W10	2/9		
		KU	2/8		
	GC	W7	2/9	23.1%	
		W10	3/9		
		KU	1/8		
	IE	W7	2/9	31.5%	
		W10	4/10		
2 **Frustration**	FF	W7	2/11	21.4%	16/73 **21.9%**
		W10	2/9		
		KU	2/8		
	GC	W7	2/9	19.2%	
		W10	2/9		
		KU	1/8		
	IE	W7	2/9	26.3%	
		W10	3/10		
3 **Unresponsiveness**	FF	W7	1/11	14.3%	13/73 **17.8%**
		W10	2/9		
		KU	1/8		
	GC	W7	2/9	19.2%	
		W10	2/9		
		KU	1/8		
	IE	W7	2/9	21%	
		W10	2/10		

Table 36: Artifacts observed in NC3 for SaaS WO

Severity Level	Nbr times observed #fails/#total			Total % of fails for SaaS WO	
1 **Annoyance**	FF	W7	3/8	59.3%	40/75 **53.3%**
		W10	6/9		
		KU	7/10		
	GC	W7	2/8	46.4%	
		W10	5/10		
		KU	6/10		
	IE	W7	11/12	55%	
		W10	0/8		
2 **Frustration**	FF	W7	0/8	29.6%	19/75 **25.3%**
		W10	3/9		
		KU	5/10		
	GC	W7	0/8	17.9%	
		W10	2/10		
		KU	3/10		
	IE	W7	6 /12	30%	
		W10	0/8		
3 **Unresponsiveness**	FF	W7	3/8	14.8%	10/75 **13.3%**
		W10	1/9		
		KU	0/10		
	GC	W7	0/8	3.6%	
		W10	1/10		
		KU	0/10		
	IE	W7	5/12	25%	
		W10	0/8		

Note that excessive latency (severity level 1) was considered existent only when more than 10 seconds latency was observed to load a large volume of data, such as the SaaS GUI and image upload to the document. This analysis was done by observing the recorded sessions against a system timer (KU) or an external timer (gadgets for W7 and W10).

The results are then summarized in Figure 46 and Figure 47 for SaaS GD and SaaS WO, respectively, in an operating-system- and browser-independent manner.

From the analysis of both charts is possible to notice that the frequency of events of severity level 1 is higher for SaaS WO (~5 times more when compared with SaaS GD), and therefore contributing to as well to an higher prevalence of events with severity level 2 (3 times more in SaaS WO than in SaaS GD). Curiously, the unavailability of the application or complete unresponsiveness, is slightly higher for SaaS GD. However, overall analysis shows that the SaaS WO seems to have more challenges under NC3 than SaaS GD. The first exhibited, during the trace collecting, more than 50% of the time QoE issues, while the latter showed QoE issues only 27% of the time. Nevertheless, both SaaS applications show a substantial degradation of the QoE in NC3.

It is worth referring that when the same study was performed for NC2, for a similar amount of collected traces, the latency was barely noticeable, and actions such as loading of the GUI of both SaaS and image had almost half the value of the latency verified under NC3 (irrespective of browser and operating system).

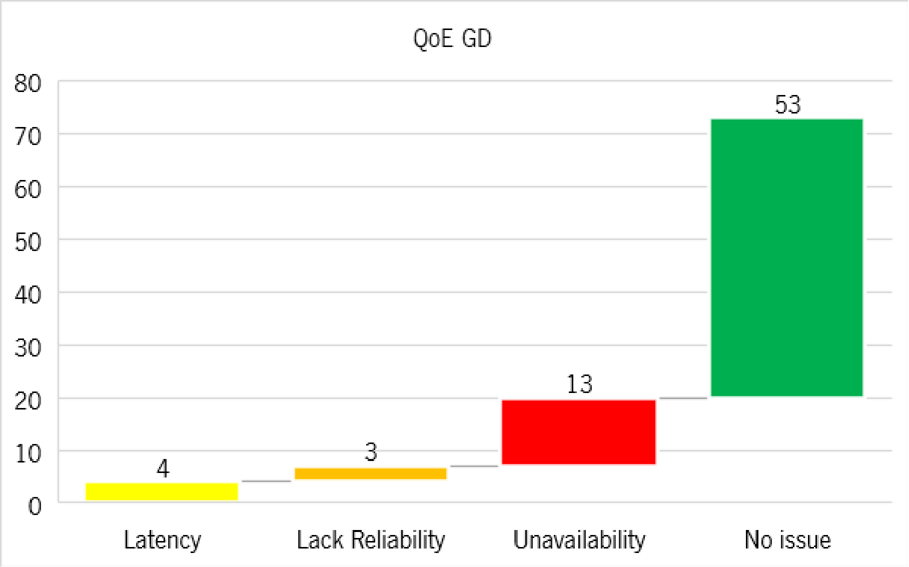


Figure 46: Perceived QoE, under NC3, for GD (irrespective of Browsers and OSs)

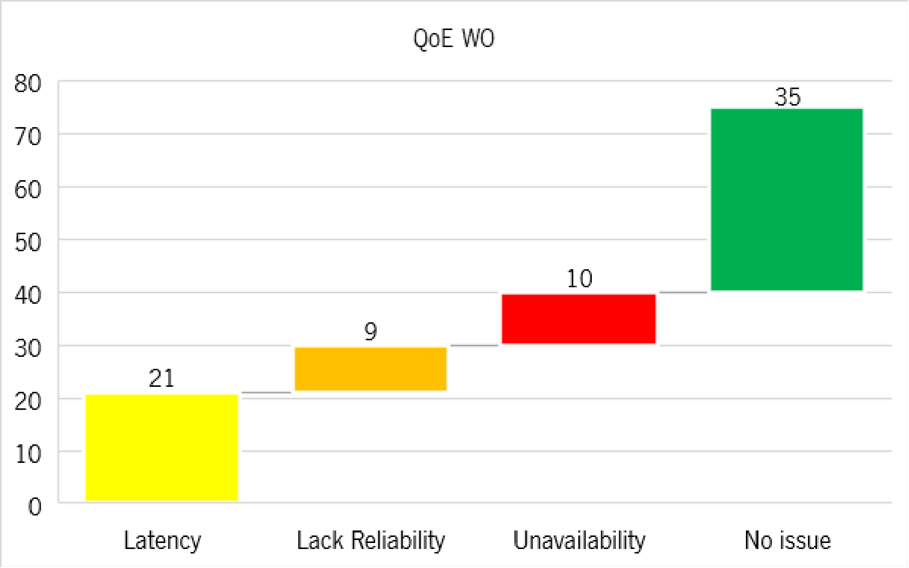


Figure 47: Perceived QoE, under NC3, for WO (irrespective of Browsers and OSs)

In conclusion, the use of SaaS with mobile and satellite-based access networks is certainly possible, though under network congestion conditions the observed QoE will likely be degraded, primarily due to increased latency, latency variation (at times, SaaS response time will become intolerable) and packet loss (at times, SaaS may simply stop responding), resulting in end user dissatisfaction and frustration, and likely – abandoning the SaaS application.

4.9 Summary of results per hypothesis

A number of hypotheses were tested in different scenarios (SaaS application, operating system (OS), browsers (BR), and network configuration (NC) combinations), with their status summarized in Table 37. Limited discussion of the observed applicability of particular hypothesis is provided, with detailed analysis included in individual sections.

Table 37: Summary of examined hypotheses

Hypothesis	Observations
A	For all examined SaaS, OS, BR, and NC combinations, Hypothesis A **holds**. A positive correlation between specific SaaS actions and peaks in traffic activity was observed, where particular network-intensive actions could be identified based on the network activity chart.
B	Hypothesis B **does not hold** when GC is used with GD, independent on NC and OS used. GC uses QUIC protocol (UDP-based), intended to replace TCP in the long run. For all other examined SaaS, OS, BR, and NC combinations, Hypothesis B **holds**, i.e., TCP is used as the exclusive transport protocol for the SaaS
C	Hypothesis C **does not hold** for GD in NC2 under KU using FF and for GD in NC3 under W10 using IE, where more traffic is downloaded than uploaded For all other examined SaaS, OS, BR, and NC combinations, Hypothesis C **holds**, with more traffic being uploaded to the SaaS than downloaded from the SaaS.
D	For all examined SaaS, OS, BR, and NC combinations, Hypothesis D **does not hold**, with OS-specific behavior observed even for the same BR, SaaS, and NC.
E	For all examined SaaS, OS, BR, and NC combinations, Hypothesis E **does not hold**, with BR-specific behavior observed even for the same OS, SaaS, and NC.
F	For all examined SaaS, OS, BR, and NC combinations, Hypothesis F **does not hold**, with NC-specific behavior observed even for the same BR, SaaS, and OS.
G	Hypotheses G **does not hold** for GD under W10, for FF, using NC3. Hypotheses G **holds partially**, with IPv6 traffic representing majority of exchanged data, for WO under W10 and under KU for all examined BR combinations. For all other examined SaaS, OS, BR, under NC3, Hypotheses G holds, with IPv6 traffic representing most of the exchanged data. The shift to IPv6 is predominantly visible in mobile networks, focusing on mobility, short-term transactional connectivity, and nomadic work.
H	Hypotheses H **does not hold**, with dramatic differences in the perceived QoE for NC3 scenarios when compared with NC1/NC2 scenarios.

Hypothesis	Observations
	High latency and packet loss observed in NC3 negatively impact the perceived QoE. The proposed QoE failure severity levels provide objective measures of issues observed in examined scenarios, correlating with the increased packet latency and packet loss in mobile networks.

5

Conclusions

The emergence of cloud-based application paradigm opens several interesting areas of study, focusing on SaaS and underlying network operation. Some of the details associated with the network operation and interaction between the SaaS host(s) and end user application rely on dynamic load balancing and cross carrier routing and, in general, can be only assessed statistically, given the constantly changing routing paths, link load, etc. Therefore, it is not possible to perform any fixed-state studies (in what concerns the routing and load balancing of the underlying network) which means that is only possible to perform a statistical analysis based in the resulting SaaS host(s) activity. This was the methodology used in this work.

The use of highly distributed and redundant SaaS infrastructures, with individual SaaS hosts providing constant state synchronization and service redundancy, provides major advantages to users looking for always-on services, with uptime guarantees, delivered anywhere in the world, and providing consistent behavior. In this work, especially in relation with Hypothesis H, shortcomings associated with bandwidth-constrained and/or high-latency access networks were identified, requiring further study to properly assess the applicability of mobile and/or satellite-based networks for this particular application paradigm.

What is even more interesting is the fact that the promised consistent SaaS behavior across different hardware and software platforms is not really delivered, given the observed differences in SaaS behavior while using different operating systems and even different browsers. Some browsers (for example, Google Chrome) may be optimized for interaction with selected SaaS platforms (for example, Google Docs), taking advantage of more streamlined communication protocols (QUIC) not observed or apparently implemented in other browsers and SaaS combinations. While this behavior is certainly a provider specific optimization which benefits a particular group of end customers, it does not seem implemented on the alternative SaaS solution from Microsoft. But it is worth note that such behavior does point for a generic trend to optimize SaaS behavior on specific hardware and/or software platforms. It is expected that this trend continues in the future, likely with new Microsoft Edge browser providing optimizations to interact with Microsoft hosted SaaS applications.

Finally, the QoE, examined in some detail in this work, does show how different aspects of the underlying transport network, as well as the SaaS applications and browser interaction affect the end-user experience. The periods of high network latency combined with packet loss and failed retransmissions turn SaaS application experience into a nightmare, where few users will find the resulting application experience superior to having a local application running on their device, allowing them to continue the work even when network connectivity is not available. In such cases, the use of a local copy of Word application (as an example), and synchronizing the document status to the cloud when and where network connectivity allows, seems to represent a better model for a highly nomadic end user, able to access network resources only sporadically. Time will tell whether the push for pervasive cloud-computing and SaaS-like applications does benefit end users, especially as mobile data networks become increasingly congested and unreliable.

The promise of future 5G networks certainly seems alluring, bringing the potential to be always on and connected, despite the fact that the initial roll outs and scarce spectrum allocation seem to provide very limited uplink capacity, required for proper operation of SaaS applications as demonstrated in this work. The focus on high-speed downloads is nothing more than constant propagation of the existing media consumption model, where server-stored and maintained content is only delivered towards the edge for consumption. Time and hardware permitting, the assessment of the SaaS performance on future 5G networks would be certainly a topic for an interesting follow-up study.

5.1 Future work

There are several areas for potential future study identified in the course of the work covered in this thesis, most of them associated closely with observations made for individual SaaS / operating system / browser combinations and their interactions. The target future study areas can be divided into the following main groups:

- SaaS application behavior consistency for more modern browsers and operating systems, adding Opera, new Microsoft Edge (based on Google Chrome), as well as other operating systems, namely Centos (different Linux branch), Mac OS, as well as some of the existing mobile device operating systems (Android, Apple). The behavior of SaaS on mobile device systems might provide interesting insights into operation of specific SaaS applications in a much more nomadic environments, with frequent network condition changes and less reliable connectivity options.

- Impact of different types of xDSL, mobile, and satellite-based access networks on SaaS operation, with the special focus on more bandwidth intensive SaaS (PO/GS) to assess the QoE for these scenarios when accessing SaaS via more bandwidth constrained and high latency access links.
- Perform a more detailed study of QoE in xDSL, mobile, and satellite-based access networks, with the special focus on network latency, packet loss, as well as such subjective aspects of QoE as SaaS usability, loss of access, loss of information, failure to save (upload) information to SaaS, etc., affecting the end-user satisfaction with the given SaaS application. The proposed study should include a test group focusing on the objective and subjective aspects of QoE assessment, validating QoE perception, assumptions, and associated observations for individuals. The study should also cover different times of the day (e.g., early morning, middle of the day, afternoon, evening) as well as different days of the week (e.g., Monday, Wednesday, Friday, and weekend), attempting to identify the impact of network bottlenecks on the SaaS QoE.
- Identify potential methods of assessing SaaS responsiveness, i.e., time delay between execution of specific actions (for example, text formatting) and the resulting SaaS response to the said change. The use of high-speed screen recording might be required in this case, to allow for sub-second delay measurements in correlation with performing specific actions in the SaaS application.
- Investigate the behavior of video conferencing application such as Hangouts, Skype, WhatsApp, etc., in the light of the findings of the office SaaS applications examined in this thesis. The real-time aspect of the video conferencing applications will present their own set of challenges as far as data capture, analysis, and also reliable reproduction of test conditions are concerned. Given the high adaptability of real-time video encoders used by such applications, new methodologies for analysis and reproduction of test conditions may need to be devised.

Bibliography

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing", 2011. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf. [Accessed Dec. 16, 2019]

[2] ISO/IEC 17788:2014, "Information technology — Cloud computing — Overview and vocabulary", 2014. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:17788:ed-1:v1:en [Accessed Dec. 16, 2019]

[3] Cisco, "Cisco Global Cloud Index: Forecast and Methodology, 2016–2021 White Paper", 2018. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html [Accessed Jan. 13, 2020]

[4] Cisco, "Cisco Annual Internet Report (2018–2023) White Paper", 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html [Accessed Feb. 12, 2020]

[5] D. Fernandes, L. Soares, J. Gomes, M. Freire and P. Inacio, "Security Issues in cloud environments: a survey" in *Int. J. of Information Security*, Vol. 13, 2014, pp. 113-170. DOI: 10.1007/s10207-013-0208-7.

[6] D. Parek and R. Sridaran, "An Analysis of Security Challenges in Cloud Computing", in *Int. J. of Advanced Computer Science and Applications*, Vol. 4, No.1, 2013, pp. 38-45. DOI: 10.14569/IJACSA.2013.040106.

[7] S. Jaydip, "Security and Privacy Issues in Cloud Computing", in *Cloud Technology: Concepts, Methodologies, Tools, and Applications,* Information Resources Management Association, USA, 2015, ch. 74, pp. 1585-1630. DOI: 10.4018/978-1-4666-6539-2.ch074.

[8] Cloud Security Alliance, "The treacherous 12 – Top Threats to Cloud Computing + Industry Insights", 2017. [Online]. Available: https://downloads.cloudsecurityalliance.org/assets/research/top-threats/treacherous-12-top-threats.pdf [Accessed Jan. 13, 2020]

[9] M. Lipp, M. Schwarz, et al., "Meltdown: Reading Kernel Memory from User Space", *in 27th USENIX Security Symposium (Security 18)*, 2018, pp. 973-990. [Online]. Available: https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-lipp.pdf [Accessed Jan. 13, 2020]

[10] P. Kocher, J. Horn, et al., “Spectre Attacks: Exploiting Speculative Execution”, in *2019 IEEE Symposium on Security and Privacy (SP),* San Francisco, CA, USA, 2019, pp. 1-19. DOI: 10.1109/SP.2019.00002.

[11] K. Ramokapane, A. Rashid and J. M. Such, “Assured Deletion in the Cloud: Requirements, Challenges and Future Directions”, in *Proceedings of the 2016 ACM on Cloud Computing Security Workshop (CCSW '16)*. Association for Computing Machinery, New York, NY, USA, 2016, pp. 97–108. DOI: 10.1145/2996429.2996434.

[12] T. Gao, P Pattabhiraman; X. Bai and W. T. Tsai, “SaaS performance and scalability evaluation in clouds.”, in *Proceedings of 2011 IEEE 6th Int. Symposium on Service Oriented System (SOSE)*, Irvine, CA, USA, 2011, pp. 61-71. DOI: 10.1109/SOSE.2011.6139093.

[13] J. Gao, et al., "A cloud-based TaaS infrastructure with tools for SaaS validation, performance and scalability evaluation.", in *4th IEEE Int. Conf. on Cloud Computing Technology and Science Proceedings*, Taipei, Taiwan, 2012, pp. 464-471. DOI: 10.1109/CloudCom.2012.6427555.

[14] S. Lehrig, R. Sanders, et al., “CloudStore — towards scalability, elasticity, and efficiency benchmarking and analysis in Cloud computing.”, in *Future Generation Computer Systems*, Vol. 78, 2018, pp. 115-126. DOI: 10.1016/j.future.2017.04.018.

[15] S. Maheshwari, D. Raychaudhuri, I. Seskar and F. Bronzino, "Scalability and Performance Evaluation of Edge Cloud Systems for Latency Constrained Applications.", in *2018 IEEE/ACM Symposium on Edge Computing (SEC),* Seattle, WA, USA, 2018, pp. 286-299. DOI: 10.1109/SEC.2018.00028.

[16] S. Aleem, F. Ahmed, R. Batool and A. Khattak, "Empirical Investigation of Key Factors for SaaS Architecture Dimension," in *IEEE Transactions on Cloud Computing*, 2019, DOI: 10.1109/TCC.2019.2906299.

[17] J. Rutkowska, “Security Challenges in Virtualized Environments”, in RSA Conference, San Franscisco, CA, USA, 2008. [Online]. Available: https://invisiblethingslab.com/resources/rsa08/Security%20Challanges%20in%20Virtualized%20Enviroments%20-%20RSA2008.pdf [Accessed Jan. 13, 2020]

[18] G. Barthe, G. Betarte, J. Campo and C. Luna, “*Cache-Leakage Resilient OS Isolation in an Idealized Model of Virtualization.*”, *in 2012 IEEE 25th Computer Security Foundations Symposium*, Cambridge, MA, USA, 2012, pp. 186-197. DOI: 10.1109/CSF.2012.17

[19] H. Moraes, R. Nunes and D. Guedes, “DCPortalsNg: Efficient Isolation of TenantNetworks in Virtualized Datacenters.”, in *Proc. of the Thirteenth International Conference on Networks,* 2014, pp 230-235.

[20] I. Odun-Ayo, S. Misra, O. Abayomi-Alli and O. Ajayi, “Cloud Multi-Tenancy: Issues and Developments.”, in *Companion Proceedings of the 10th International Conference on Utility and Cloud Computing (UCC '17 Companion)*, Association for Computing Machinery, New York, NY, USA, 2017, pp. 209–214. DOI: 10.1145/3147234.3148095.

[21] W. Tsai, X. Bai and Y. Huang, “Software-as-a-service (SaaS): Perspectives and challenges.”, in Science China – Information Sciences, Vol. 57, 2014, pp. 1-15. DOI: 10.1007/s11432-013-5050-z.

[22] Z. Yang, J. Sun, Y. Zhang and Y. Wang, “Understanding SaaS adoption from the perspective of organizational users: A tripod readiness model.”, in *Computers in Human Behavior*, Vol. 45, 2015, pp. 254-264. DOI: 10.1016/j.chb.2014.12.022.

[23] X. Tan and Y. Kim, "User acceptance of SaaS-based collaboration tools: a case of Google Docs.", in *Journal of Enterprise Information Management*, Vol. 28, No. 3, 2015, pp. 423-442. DOI 10.1108/JEIM-04-2014-0039.

[24] L. Dinh-Xuan, C. Schwartz, et al., “Analyzing the Impact of Delay and Packet Loss on Google Docs.”, in *7th International Conference on Mobile Networks and Management*, MONAMI 2015. LNCSSITE – Vol. 158, R. Agüero, T. Zinner, M. García-Lozano, BL Wenning, A. Timm-Giel. (eds), Cham: Springer, 2015, pp. 211-224. DOI: 10.1007/978-3-319-26925-2_16.

[25] D. Zissis and D. Lekkas, “Addressing cloud computing security issues.”, in *Future Generation Computer Systems,* Vol. 28, No. 3, 2012, pp. 583-592. DOI: 10.1016/j.future.2010.12.006

[26] N. Khan and A. Al-Yasiri, “Cloud Security Threats and Techniques to Strengthen Cloud Computing Adoption Framework.*”,* in *Cyber Security and Threats: Concepts, Methodologies, Tools, and Applications,* Information Resources Management Association, USA, 2018, ch. 16, pp. 268-285. DOI: 10.4018/978-1-5225-5634-3.ch016.

[27] G. Aceto, A. Botta, W. Donato and A. Pescapè, “Cloud monitoring: A survey”, in Computer Networks, Vol. 57, No. 9, 2013, pp. 2093-2115 2013. DOI: 10.1016/j.comnet.2013.04.001.

[28] N. Palhares, S. Rito Lima and P. Carvalho, “A Multidimensional Model for Monitoring Cloud Services”, in *Advances in Information Systems and Technologies*”, AISC – Vol. 206, Á. Rocha, A. Correia, T. Wilson, K. Stroetmann (eds), Berlin: Springer, 2013, pp. 931-938. DOI: 10.1007/978-3-642-36981-0_87.

[29] I. Drago, M. Mellia, et al, "Inside dropbox: understanding personal cloud storage services.*", in Proc. of the 2012 Internet Measurement Conference (IMC '12)*, Association for Computing Machinery, New York, NY, USA, 2012, pp. 481-494. DOI: 10.1145/2398776.2398827.

[30] D. Oliveira, P. Carvalho and S. Lima, "Understanding Cloud Storage Services Usage: A Practical Case Study", in *International Conference on Networked Systems*, NETYS 2015, LNCS – Vol. 9466, A. Bouajjani, H. Fauconnier. (eds), Cham: Springer, Cham, 2015, p. 501-506. 2015. DOI: 10.1007/978-3-319-26850-7_39.

[31] P. Casas, H. Fischer, et al., "A first look at quality of experience in Personal Cloud Storage services*", in 2013 IEEE International Conference on Communications Workshops (ICC).* Budapest: IEEE, 2013, pp. 733-737. DOI: 10.1109/ICCW.2013.6649330.

[32] S. Al-Shammari, A. Al-Yasiri, "Defining a Metric for Measuring QoE of SaaS Cloud Computing", in *Proc. of the 15th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking Broadcasting (PGNET'14)*, UK, 2014, pp. 251-256. [Online]. Available:https://www.researchgate.net/publication/264548274_Defining_a_Metric_for_Measuring_QoE_of_SaaS_Cloud_Computing [Accessed Feb. 12, 2020]

[33] T. Hobfeld, R. Schatz, M. Varela, C. Timmerer, "Challenges of QoE management for cloud applications", in *IEEE Communications Magazine*, Vol. 50, no. 4, 2012, pp. 28-36. DOI: 10.1109/MCOM.2012.6178831.

[34] Y-R Shin and E-N Huh, "mCSQAM: Service Quality Assessment Model in Mobile Cloud Services Environment" in *Mobile Information Systems*, 2016, pp. 1-9. DOI: 10.1155/2016/2517052.

[35] Y. Rekhter, B. Moskowitz, et al., "RFC1918: Address Allocation for Private Internets", 1996. [Online]. Available: https://dl.acm.org/doi/pdf/10.17487/RFC1918 [Accessed Feb. 12, 2020]

[36] J. Han, M. Kamber and J Pei, "Data Preprocessing" in *Data Mining: Concepts and Techniques,* 3rd ed., ch. 3, Waltham, MA, USA: Morgan Kaufmann, 2011, pp. 83 – 123. ISBN: 978-0-12-381479-1.

[37] M. Seufert, R. Schatz, et al., "Is QUIC becoming the New TCP? On the Potential Impact of a New Protocol on Networked Multimedia QoE", in *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, Berlin, Germany: IEEE, 2019, pp. 1-6. DOI: 10.1109/QoMEX.2019.8743223.

[38] T. Brisco, "RFC1794: DNS Support for Load Balancing", 1995. [Online]. Available: https://dl.acm.org/doi/pdf/10.17487/RFC1794 [Accessed Feb. 12, 2020]

[39] S. Narayan, Y. Shi, "TCP/UDP Network Performance Analysis of Windows Operating Systems with IPv4 and IPv6", in *2010 2nd International Conference on Signal Processing Systems (ICSPS)*, Dalian, China: IEEE, 2010, pp. V2-219-V2-222. DOI: 10.1109/ICSPS.2010.5555285.

[40] R. Draves, "RFC3484: Default Address Selection for Internet Protocol version 6 (IPv6)", 2003. [Online]. Available: https://dl.acm.org/doi/pdf/10.17487/RFC3484 [Accessed Feb. 12, 2020]

[41] Internet Society, "State of IPv6 Deployment 2018", 2018. [Online]. Available: https://www.internetsociety.org/wp-content/uploads/2018/06/2018-ISOC-Report-IPv6-Deployment.pdf [Accessed Feb. 12, 2020]

[42] M. Blanchet, "*Migrating to IPv6: A Practical Guide to Implementing IPv6 in Mobile and Fixed Networks*", Québec, Canada:John Wiley & Sons, Ltd, 2009. ISBN: 9780470468906.

[43] A. Conta and S. Deering, "RFC2473: Generic Packet Tunneling in IPv6 Specification", 1998. [Online]. Available: https://dl.acm.org/doi/pdf/10.17487/RFC2473 [Accessed Feb. 12, 2020]

[44] ITU-T Recommendation P.10, "Vocabulary for performance, quality of service and quality of experience", 2017. [Online]. Available: https://www.itu.int/rec/T-REC-P.10 [Accessed Dec. 18, 2019]

[45] "Qualinet White Paper on Definitions of Quality of Experience" in European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), P. Callet, S. Möller and A. Perkis (eds.), Lausanne, Switzerland, Version 1.2, 2013. [Online]. Available: http://www.qualinet.eu/images/stories/QoE_whitepaper_v1.2.pdf [Accessed Dec. 18, 2019]

APPENDIX A – WIRESHARK PACKET FILTERS

GD scenarios in NC1 network configuration

```
(!stp && !dns && !ocsp && !arp && !smb && !smb2 && !dhcpv6 && !nbns
&& !snmp && !llmnr && !nbss && !bootp && !icmp && !icmpv6 && !ipv6 &&
! (udp.dstport == 1900) && !(ip.src == 192.168.1.125 && ip.dst >=
192.168.1.1 && ip.dst <= 192.168.1.255) && !(ip.dst == 192.168.1.125
&& ip.src >= 192.168.1.1 && ip.src <= 192.168.1.255) && !(ip.dst >=
224.0.0.0 && ip.dst<= 239.255.255.255)) && (ip.addr == 64.233.160.0/19
|| ip.addr == 216.58.192.0/19 || ip.addr == 173.194.0.0/16 || ip.addr
== 172.217.0.0/16)
```

GD scenarios in NC2 network configurations

```
(!stp && !dns && !ocsp && !arp && !smb && !smb2 && !dhcpv6 && !nbns
&& !snmp && !llmnr && !nbss && !bootp && !icmp && !icmpv6 && !ipv6 &&
! (udp.dstport == 1900) && !(ip.src == 192.168.1.125 && ip.dst >=
192.168.1.1 && ip.dst <= 192.168.1.255) && !(ip.dst == 192.168.1.125
&& ip.src >= 192.168.1.1 && ip.src <= 192.168.1.255) && !(ip.dst >=
224.0.0.0 && ip.dst<= 239.255.255.255) && !ip.addr == 192.168.1.255
&& !ip.addr == 255.255.255.255 && !ip.addr == 192.168.1.200 &&
!ip.addr == 192.168.1.199 && !ip.addr == 192.168.1.203 && !ip.addr ==
192.168.1.1 && !ip.addr == 192.168.56.1 && !ip.addr == 192.168.1.122
&& !ip.addr == 192.168.1.136 && !ip.addr == 192.168.1.130) && (ip.addr
== 172.217.0.0/16 || ip.addr == 74.125.0.0/16 || ip.addr ==
173.194.0.0/16 || ip.addr == 216.239.32.0/19 || ip.addr ==
216.58.192.0/19 || ip.addr == 209.85.128.0/17 || ip.addr ==
108.177.0.0/17)
```

GD scenarios in NC3 network configuration

```
(!stp && !dns && !ocsp && !arp && !smb && !smb2 && !dhcpv6 && !nbns
&& !snmp && !llmnr && !nbss && !bootp && !icmp && !icmpv6 && !
(udp.dstport == 1900) && !(ip.dst >= 224.0.0.0 && ip.dst<=
239.255.255.255) && !ip.addr == 192.168.1.255 && !ip.addr ==
255.255.255.255) && (ip.addr == 216.58.192.0/19 || ip.addr ==
172.217.0.0/16 || ip.addr == 216.239.32.0/19 || ip.addr ==
209.85.128.0/17 || ipv6.addr == 2607:f8b0::/32 || ip.addr ==
34.64.0.0/10 || ip.addr == 74.125.0.0/16)
```

WO scenarios in NC2 network configuration

```
(!stp && !dns && !ocsp && !arp && !smb && !smb2 && !dhcpv6 && !nbns
&& !snmp && !llmnr && !nbss && !bootp && !icmp && !icmpv6 && !ipv6 &&
! (udp.dstport == 1900) && !(ip.src == 192.168.1.125 && ip.dst >=
192.168.1.1 && ip.dst <= 192.168.1.255) && !(ip.dst == 192.168.1.125
&& ip.src >= 192.168.1.1 && ip.src <= 192.168.1.255) && !(ip.dst >=
224.0.0.0 && ip.dst<= 239.255.255.255) && !ip.addr == 192.168.1.255
&& !ip.addr == 255.255.255.255 && !ip.addr == 192.168.1.200 &&
!ip.addr == 192.168.1.199 && !ip.addr == 192.168.1.203 && !ip.addr ==
192.168.1.1 && !ip.addr == 192.168.56.1 && !ip.addr == 192.168.1.122)
```

```
&& (ip.addr == 52.96.0.0/12 || ip.addr == 40.80.0.0/12 || ip.addr ==
204.79.196.0/23 || ip.addr == 20.180.0.0/14 || ip.addr == 13.96.0.0/13
|| ip.addr == 52.224.0.0/11 || ip.addr == 40.76.0.0/14 || ip.addr ==
204.79.195.0/24 || ip.addr == 20.128.0.0/16 || ip.addr == 13.64.0.0/11
|| ip.addr == 52.160.0.0/11 || ip.addr == 40.74.0.0/15 || ip.addr ==
20.64.0.0/10 || ip.addr == 157.60.0.0/16 || ip.addr == 13.104.0.0/14
|| ip.addr == 52.152.0.0/13 || ip.addr == 40.64.0.0/13 || ip.addr ==
20.48.0.0/12 || ip.addr == 157.56.0.0/14 || ip.addr == 104.40.0.0/13
|| ip.addr == 52.148.0.0/14 || ip.addr == 40.125.0.0/17 || ip.addr ==
20.40.0.0/13 || ip.addr == 157.54.0.0/15 || ip.addr == 104.208.0.0/13
|| ip.addr == 52.146.0.0/15 || ip.addr == 40.124.0.0/16 || ip.addr ==
20.36.0.0/14 || ip.addr == 137.135.0.0/16 || ip.addr == 52.145.0.0/16
|| ip.addr == 40.120.0.0/14 || ip.addr == 20.34.0.0/15 || ip.addr ==
131.253.18.0/24 || ip.addr == 52.112.0.0/14 || ip.addr ==
40.112.0.0/13 || ip.addr == 20.33.0.0/16 || ip.addr == 131.253.16.0/23
|| ip.addr == 40.96.0.0/12 || ip.addr == 23.96.0.0/13 || ip.addr ==
20.184.0.0/13 || ip.addr == 131.253.12.0/22)
```

WO scenarios in NC3 network configuration

```
(!stp && !dns && !ocsp && !arp && !smb && !smb2 && !dhcpv6 && !nbns
&& !snmp && !llmnr && !nbss && !bootp && !icmp && !icmpv6 && !
(udp.dstport == 1900) && !(ip.dst >= 224.0.0.0 && ip.dst<=
239.255.255.255) && !ip.addr == 192.168.1.255 && !ip.addr ==
255.255.255.255) && (ip.addr == 52.96.0.0/12 || ip.addr ==
52.112.0.0/14 || ip.addr == 51.140.0.0/14 || ip.addr == 40.96.0.0/12
|| ip.addr == 40.80.0.0/12 || ip.addr == 40.76.0.0/14 || ip.addr ==
40.74.0.0/15 || ip.addr == 40.125.0.0/17 || ip.addr == 40.124.0.0/16
|| ip.addr == 40.120.0.0/14 || ip.addr == 40.112.0.0/13 || ipv6.addr
== 2620:1EC::/36 || ip.addr == 20.64.0.0/10 || ip.addr == 20.48.0.0/12
|| ip.addr == 20.40.0.0/13 || ip.addr == 20.36.0.0/14 || ip.addr ==
20.34.0.0/15 || ip.addr == 20.33.0.0/16 || ip.addr == 20.128.0.0/16
|| ip.addr == 157.60.0.0/16 || ip.addr == 157.56.0.0/14 || ip.addr ==
157.54.0.0/15 || ip.addr == 131.253.18.0/24 || ip.addr ==
131.253.16.0/23 || ip.addr == 131.253.12.0/22 || ip.addr ==
13.96.0.0/13 || ip.addr == 13.64.0.0/11 || ip.addr == 13.104.0.0/14)
```

Appendix B – List of SaaS host addresses per examined scenario

Table 38: List of SaaS host subnets in NC1 GD scenarios

64.233.160.0/19	216.58.192.0/19	173.194.0.0/16	172.217.0.0/16

Table 39: List of IP subnets excluded from NC2 GD scenarios

97.112.0.0/12	8.224.0.0/11	54.64.0.0/13	54.144.0.0/12	52.32.0.0/11
52.0.0.0/11	13.224.0.0/14	104.16.0.0/12		

Table 40: List of SaaS host subnets in NC2 GD scenarios

172.217.0.0/16	74.125.0.0/16	173.194.0.0/16	216.239.32.0/19
216.58.192.0/19	209.85.128.0/17	108.177.0.0/17	

Table 41: List of IP subnets excluded from NC2 GD scenarios

99.87.128.0/18	52.96.0.0/12	40.80.0.0/12	20.64.0.0/10	134.170.0.0/16
99.87.0.0/17	52.32.0.0/11	40.76.0.0/14	20.48.0.0/12	13.96.0.0/13
99.86.0.0/16	52.224.0.0/11	40.74.0.0/15	20.40.0.0/13	13.64.0.0/11
99.85.128.0/17	52.160.0.0/11	40.64.0.0/13	20.36.0.0/14	13.248.0.0/14
99.84.0.0/16	52.152.0.0/13	40.125.0.0/17	20.34.0.0/15	13.224.0.0/14
99.83.64.0/18	52.148.0.0/14	40.124.0.0/16	20.33.0.0/16	13.104.0.0/14
99.83.128.0/17	52.146.0.0/15	40.120.0.0/14	20.128.0.0/16	104.64.0.0/10
96.16.0.0/15	52.145.0.0/16	40.112.0.0/13	184.84.0.0/14	104.40.0.0/13
8.224.0.0/11	52.112.0.0/14	23.64.0.0/14	157.60.0.0/16	100.24.0.0/13
74.125.0.0/16	52.0.0.0/11	23.32.0.0/11	157.56.0.0/14	100.20.0.0/14
72.21.80.0/20	45.60.0.0/16	23.192.0.0/11	157.54.0.0/15	
54.64.0.0/13	40.96.0.0/12	23.0.0.0/12	143.204.0.0/16	

Table 42: List of SaaS host subnets in NC2, WO scenarios

52.96.0.0/12	40.80.0.0/12	204.79.196.0/23	20.180.0.0/14	13.96.0.0/13
52.224.0.0/11	40.76.0.0/14	204.79.195.0/24	20.128.0.0/16	13.64.0.0/11
52.160.0.0/11	40.74.0.0/15	20.64.0.0/10	157.60.0.0/16	13.104.0.0/14
52.152.0.0/13	40.64.0.0/13	20.48.0.0/12	157.56.0.0/14	104.40.0.0/13
52.148.0.0/14	40.125.0.0/17	20.40.0.0/13	157.54.0.0/15	104.208.0.0/13
52.146.0.0/15	40.124.0.0/16	20.36.0.0/14	137.135.0.0/16	
52.145.0.0/16	40.120.0.0/14	20.34.0.0/15	131.253.18.0/24	
52.112.0.0/14	40.112.0.0/13	20.33.0.0/16	131.253.16.0/23	
40.96.0.0/12	23.96.0.0/13	20.184.0.0/13	131.253.12.0/22	

Table 43: List of IP addresses excluded from NC2 WO scenarios

96.6.0.0/15	45.60.0.0/16	23.64.0.0/14	173.194.0.0/16	13.224.0.0/14
74.125.0.0/16	35.176.0.0/13	23.32.0.0/11	172.224.0.0/12	104.64.0.0/10
72.21.80.0/20	35.160.0.0/12	23.192.0.0/11	172.217.0.0/16	
67.128.0.0/13	35.152.0.0/13	23.0.0.0/12	152.192.0.0/13	
52.64.0.0/12	34.192.0.0/10	205.185.192.0/19	152.176.0.0/12	
52.0.0.0/11	3.208.0.0/12	184.84.0.0/14	151.101.0.0/16	

Table 44: List of SaaS host subnets in NC3 GD scenarios

216.58.192.0/19	172.217.0.0/16	216.239.32.0/19	209.85.128.0/17
2607:f8b0::/32	34.64.0.0/10	74.125.0.0/16	

Table 45: List of IP subnets excluded from NC3 GD scenarios

2607:fb90:6c29:5d75:99bc::/80	64::/16	2a00:1450:4019:801::/64	2600:9000::/28
54.64.0.0/13	52.64.0.0/12	52.32.0.0/11	52.0.0.0/11
34.192.0.0/10	13.224.0.0/14	184.84.0.0/14	91.189.88.0/20

Table 46: List of SaaS host subnets in NC3 WO scenarios

52.96.0.0/12	40.74.0.0/15	20.64.0.0/10	20.128.0.0/16	131.253.12.0/22
52.112.0.0/14	40.125.0.0/17	20.48.0.0/12	157.60.0.0/16	13.96.0.0/13
51.140.0.0/14	40.124.0.0/16	20.40.0.0/13	157.56.0.0/14	13.64.0.0/11
40.96.0.0/12	40.120.0.0/14	20.36.0.0/14	157.54.0.0/15	13.104.0.0/14
40.80.0.0/12	40.112.0.0/13	20.34.0.0/15	131.253.18.0/24	
40.76.0.0/14	2620:1EC::/36	20.33.0.0/16	131.253.16.0/23	

Table 47: List of IP subnets excluded from NC3 WO scenarios

ff02::/16	52.64.0.0/12	2606:4700::/32	216.58.192.0/19	151.101.0.0/16
96.6.0.0/15	52.32.0.0/11	2600:9000::/28	198.105.240.0/20	143.204.0.0/16
96.16.0.0/15	52.0.0.0/11	2600:1400::/24	192.184.64.0/20	104.64.0.0/10
72.246.0.0/15	2a03:2880::/29	23.64.0.0/14	192.168.0.0/16	104.244.40.0/21
72.21.80.0/20	2620:118:7000::/44	23.32.0.0/11	184.24.0.0/13	104.16.0.0/12
72.21.192.0/19	2607:FB90::/28	23.192.0.0/11	172.217.0.0/16	
64:ff9b::/32	2607:F8B0::/32	23.111.8.0/22	152.192.0.0/13	
54.144.0.0/12	2607:7700::/32	23.0.0.0/12	152.176.0.0/12	