ISP Traffic Management via Flow Optimization

vorgelegt von

Dipl.-Ing.

Emmanuel Obi Akonjang

geb. in Victoria

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr. habil. Thomas Zinner, Technische Universität Berlin, Germany
Gutachter:	Prof. Georgios Smaragdakis, Ph. D., Technische Universität Berlin, Germany
Gutachterin:	Prof. Anja Feldmann, Ph. D., Max Planck Institute for Informatics, Germany
Gutachter:	Prof. Dr. Olaf Maennel, Tallinn University of Technology, Estonia

Tag der wissenschaftlichen Aussprache: 21.12.2018

Berlin 2019

Eidesstattliche Erklärung

Ich versichere an Eides statt, dass ich diese Dissertation selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Datum Emmanuel Obi Akonjang (Dipl.-Ing.)

Abstract

A major challenge that Internet Service Providers (ISPs) face today, is the growing traffic volume that they must handle. The situation becomes even more complex, as users' demands become more volatile or when new application trends emerge that fundamentally change traffic dynamics and composition. Thus, ISPs are always compelled to deal with such changes, in a proper, efficient, and timely manner.

One typical example of such a trend, which has a profound effect on traffic volume, traffic composition, traffic dynamics, overall network performance and user experience, is Peer-to-Peer (P2P). P2P's disruptive and high bandwidth-consuming nature, poses a number of challenges to an ISP's capability to effectively manage traffic on its own network. This is a major concern/issue that calls for quick and effective mitigation approaches by ISPs. The mitigation process starts with studying and understanding P2P systems and their protocols.

Studies reveal a number of important characteristics and shortcomings of P2P systems, including: i) high churn, ii) selfish construction and maintenance of the P2P overlay, iii) unawareness of the underlying network conditions, iv) mismatch between P2P overlay and ISP underlay, v) high management traffic, vi) high proportion of cross-AS neighbor-relationships and cross-AS traffic, vii) a data flow between two peers often crosses AS boundaries multiple times.

Based on the above, in this thesis, we propose a solution that promotes collaboration between the P2P overlay and the ISP underlay, leading to a win-win situation for both parties. In detail, we propose an ISP-operated value-added service that we call the *Oracle*. The Oracle provides an interface for peers to become aware of network conditions, thus improve peer selection and performance of P2P applications. The Oracle does this by sorting a peer's list of potential neighbors/download sources to favor locality, i.e. peers that are in the same AS domain and even in the same geographical location like the requesting peer. With the Oracle, we show that improved performance that benefits ISPs, applications and users are attainable.

We conduct a series of packet-level simulations to study and quantify potential gains offered by the Oracle service. We start with an implementation of the Gnutella P2P protocol within the SSFNet simulation environment. We next model different ISP and P2P topologies, to study the effects of the Oracle across diverse topologies, then use various mathematical distributions to model user behavioral patterns that reflect realistic, inauspicious and best-case scenarios. Using appropriate metrics, we quantify and compare the performance between unbiased (no use of the Oracle service) and biased (use of the Oracle service) topologies. In nearly all categories, our analyses reveal superior performances for Oracle-biased topologies.

We next shift our focus to backbone networks and study how their topologies affect general network and application performance in the presence of normal and heavy traffic load. For this, we use a reference backbone network model for Germany

and derive 3 different topologies from it. By keeping the number of nodes (PoPs) in the topologies constant at 12 and varying the number (and size) of their links from 66 (fullmesh), to 30 and 20 respectively, we obtain 3 topologies that differ in capacity (total bandwidth) and other topological/structural properties. We use selected metrics to assess and compare their performance under the same traffic conditions. Our analyses show similarities as well as differences in performance between the topologies. We observe that, for a few categories at baseline traffic, the performance in the 20 links topology (which has the smallest number of links and the least total bandwidth) is comparable to those in the other 2 topologies. However, when the traffic increases by 35%, the performance in the 20 links topology worsens and becomes the least in nearly all compared categories.

A deeper analysis of the 20 links topology reveals that, while some major (high bandwidth) links are suffering from congestion, other links, mostly of lesser bandwidth, have little to no traffic on them. Instead of upgrading the congested links, like most ISPs would normally opt to do, we propose a more efficient and cost effective solution. It involves the use of mathematical optimization to influence traffic flows and achieve better network and application performance. We argue and show that better performance is attainable by minimizing the maximum link utilization and efficiently distributing the traffic load across all links in the topology. Our main objectives are congestion avoidance in the ISP network, performance enhancement for applications that run on the ISP network and better cost control by the ISP.

Zusammenfassung

Eine große Herausforderung für Internetdienstanbietern (ISPs) im Internet, ist die wachsende Menge an Verkehr, die sie stemmen müssen. Die Lage wird komplexer, als die Nachfrage der Benutzer/Kunden volatiler wird oder wenn populäre Anwendungstrends entstehen, die Dynamik und Komposition des Verkehrs fundamental verändern. ISPs sehen sich gezwungen mit solchen Veränderungen in angemessener, effizienter und zeitgemäßer Art und Weise umzugehen.

Ein typisches Beispiel eines solchen Trends, welcher maßgebliche Auswirkungen auf Menge, Komposition, und Dynamik von Verkehr, Leistungsfähigkeit des Netzwerks, und User Experience hat, ist Peer-to-Peer (P2P). Die disruptive und hohe Bandbreitenkonsumierende Natur von P2P stellt die Fähigkeit von ISPs, das eigene Netzwerk zu managen, vor einige Herausforderungen. Es stellt eine Hauptsorge der ISPs dar und erforderte schnelle und effektive Abmilderungsansätze. Solche Abmilderungsprozesse beginnen mit der Untersuchung und dem Verständnis von P2P-Systemen und ihrer Protokolle.

Studien bringen einige wichtige Charakteristiken und Defizite von P2P-Systemen hervor, unter anderem: i) hohen Churn. ii) eigennützige Errichtung und Wartung des P2P Overlays iii) Unwissenheit des Underlay Netzwerkzustandes iv) Diskrepanz wegen mangelnden Zusammenhang zwischen dem P2P-Overlay und dem Underlay des ISPs v) hoher Signalingoverhead, die zu zusätzlichen Verkehr führt. vi) hoher Anteil an Cross-AS-Nachbarschaftsbeziehungen und Cross-AS-Verkehr vii) oft, der Datenfluss zwischen einer Quelle und einem Ziel überquert mehrmals die AS-Grenzen.

Basierend auf diesen Fakten, schlagen wir eine Lösung vor, die die Kollaboration zwischen P2P-Overlay und ISP-Underlay fördert, und beiden Seiten Vorteile bringt. Wir schlagen einen vom ISP betriebenen, Mehrwert-Service vor, den wir *Oracle* nennen. Das Oracle bietet eine Schnittstelle, womit Peers, den Zustand des Netzwerks erkundigen können, um eine Verbesserung der Peer-Auswahl und der Performance von P2P-Anwendungen zu ermöglichen. Dabei sortiert es eine Liste von potentiellen Nachbarn/Downloadquellen der Peers, basierend auf deren örtlichen Lage, d.h., Peers die sich in der selben Domain und in der selben geografischen Lage befinden, wie der anfordernde Peer. Unser Hauptziel ist es die oben genannten Probleme abzumildern und zugleich dem ISP und den P2P-Benutzern Vorteile zu bieten.

Wir führen mehrere Packet-Level-Simulationen durch, um die potentiellen Vorteile, die das Oracle zu bieten hat, zu studieren und zu quantifizieren. Wir beginnen mit einer Implementierung des Gnutella P2P-Protokolls innerhalb der SSFNet Simulationsumgebung. Als nächstes modellieren wir verschiedene ISP- und P2P-Topologien, um die Auswirkungen des Oracles auf unterschiedliche Topologien zu studieren, gefolgt von der Modellierung von Benutzerverhalten, die realistische, ungünstige und best-case Szenarien, mittels verschiedener mathematischer Verteilungen, darstellen. Wir quantifizieren und vergleichen die Leistung von nicht-modifizierten (oh-

ne Oracle-Service) und modifizierten (mit Oracle-Service) Topologien. In fast allen Kategorien bringen unsere Analysen eine erhöhte Leistung bei Topologien mit Oracle-Unterstützung hervor.

Als nächstes, setzen wir unseren Fokus auf Backbone-Netzwerke und analysieren, wie deren Topologien sich auf die allgemeine Netzwerk- und Applikationsleistung, in Gegenwart von normaler und starker Verkehrslast, auswirken. Dazu verwenden wir ein Referenznetzwerkmodell für Deutschland und leiten daraus 3 unterschiedliche Topologien ab. Während wir die Anzahl der Knoten (PoPs) in den Topologien konstant bei 12 halten, und die Anzahl (und Größe) ihrer Links variieren, also zwischen 66 (fullmesh), 30 und 20, erhalten wir 3 Topologien, die sich in Kapazität (gesamte Bandbreite) und anderen topologischen/strukturellen Eigenschaften unterscheiden. Wir verwenden ausgewählte Metriken um die Leistung unter gleichen Verkehrsbedingungen zu ermitteln und zu vergleichen. Unsere Analysen zeigen Ähnlichkeiten und Unterschiede in der Leistung zwischen den Topologien. Wir beobachten, dass die Leistung in der 20-Links-Topologie (welche die wenigstens Links und die kleinste Gesamtbandbreite besitzt) nur in einiger weniger Kategorien vergleichbar ist mit denen der anderen beiden Topologien. Und zwar, nur bei Baseline-Verkehr. Bei Erhöhung des Verkehrs um 35%, verringert sich die Leistung in der 20-Links-Topologie in fast allen Kategorien zur geringsten.

Eine genauere Analyse der 20-Links-Topologie zeigt, dass während einige große Links (mit hohen Bandbreite) von Verkehrsstau beeinträchtigt sind, andere Links, meist mit weniger Bandbreite, wenig bis kein Verkehr haben. Anstatt die beeinträchtigen Links auszubauen, wie die meisten ISPs es tun würden, empfehlen wir eine effizientere und kosteneffektivere Lösung. Es beinhaltet die Nutzung mathematischer Optimierungen zur Beeinflussung des Verkehrs und der Realisierung besserer Leistung. Wir glauben und zeigen, dass die Verbesserung der Leistung, durch Minimierung der maximalen Auslastung der Links und effizienter Verteilung der Verkehrslast über alle Links in der Topologie, erreicht werden kann. Unsere Hauptziele sind Vermeidung von Verkehrsstau im ISP-Netzwerk, Leistungssteigerung für Anwendungen, die das ISP-Netzwerk verwenden und bessere Kostenkontrolle durch den ISP.

Acknowledgments

Whatever you do [no matter what it is] in word or deed, do everything in the name of the Lord Jesus [and in dependence on Him], giving thanks to God the Father through Him.

Colossians 3:17 (AMP)

All things are possible with God. I thank Him for my life and for His divine intervention in my challenges and battles, especially during the period of this project. This thesis would not have been possible, if not of the people He sent to guide, advice, encourage and support me.

First and foremost, I would like to specially and sincerely thank my advisor, mentor and guide, Prof. Anja Feldmann, for her relentless support, inspiring technical advice, constant encouragement and great patience throughout the duration of this project. She always made time for us to discuss my progress, pointing me to the right ideas, relevant literature and people, whenever I had needed them (which was *always*). I am very grateful to have learned and richly benefited from her methodical and practical approach to research and her attention to details. Her easygoing nature is simply exemplary and very inspiring. I really can not thank her enough for all she has done and the role she has played in my life.

I would also like to heartily thank Prof. Georgios Smaragdakis, the co-advisor and my mentor, for the uncountable hours of technical and non-technical advice, discussions and encouragements that he has offered me over the years. I have benefited a lot from his deep insight and expertise. He always made time for me, even when he was pressed with other important activities. I am very grateful for all his help.

My thanks also go to Prof. Steve Ulig for the initial discussions we had on backbone networks and network structural properties, back in his T-Labs days and to Prof. Thomas Zinner for the discussions and advice he offered at the latter stage of this project.

Many thanks to Prof. Olaf Maennel for accepting to be an examiner of this thesis and to all the members of the PhD committee for their precious time spent in reviewing my work.

I've had a wonderful time, working with former and current members of the Intelligent Networking (FG-INET) team. I am very grateful for all the discussions, ideas, objective criticisms, help, contributions and lessons that each of them has offered. I especially thank Vinay Aggarwal, whom I worked closely with and whose brilliant contributions boosted our progress, Matthias Rost, for his invaluable insight, help and discussions on mathematical programming, Thomas Krenc, for the many discussions and support, as well as his great help with translation into perfect

German. Many thanks also to Ingmar Poese and Benjamin Frank, great colleagues, with whom I had the privilege and joy to work on some projects.

I'm also grateful to the former and current Administrative Assistants, Britta Schneider and Birgit Hohmeier-Touré, respectively for their ceaseless readiness to help, especially with bureaucratic follow-ups. My thanks also go to Rainer May, our able System Administrator, who continuously ensures systems remain up and running. He is always ready to help out when things go wrong or when we mess them up. Much thanks also to Sarah Dierenfeld, for her patience and help, in sorting out the issues I occasionally had with my account or access to particular systems.

In no particular order, I would also like to extend special thanks to some close relatives and friends; Serge Ngueda, Dr. Ivan Ndip, Bigoh Akonjang, Bert Salz and Walter Agbor Bawa (PharmD), for their encouragement, support and prayers over the years. I will always remain grateful to them, especially for their moral and spiritual support.

Sure and of course, I would as well like to thank my wife, Oneke and our children, Agbor-Toko, Babbey, Mikaili and Sarahila, for their endless love, patience, support and sacrifices. I really appreciate their understanding, especially whenever I was quite pressed and repeatedly couldn't make time to join them for family activities, trips and vacations.

Last but not least, my apologies to all those whose names I could not explicitly mention. Your contributions will never be forgotten and I will always remain thankful and grateful to you as well.

Dedication

In dedication to my late mum, Frida Agbor-Toko Akonjang.

Thank you mum for the love you showed me from the day I was born, the lessons you taught me, the core values you instilled in me as a child, your ceaseless encouragements and your advice to take on and complete this project. Unfortunately, your long battle with cancer started just when this project was gathering speed. The short up-phases and long down-phases that ensued, your strong will to keep on fighting, my restlessness and worries in seeing you go through so much pains, then the devastation brought by the news of your passing away ... no day goes by without me thinking of you, mum.

I will always miss you.

Publications

Conferences and Workshops

Ingmar Poese, Obi Akonjang, Anja Feldmann, and Georgios Smaragdakis
Implementation of a Proxidor Use Case.
Talk at IETF-74, San Francisco, USA, March 2009

Vinay Aggarwal, Obi Akonjang and Anja Feldmann
ISP-Aided Neighbor Selection in P2P Systems
Internet Engineering Task Force (IETF) P2P Infrastructure Workshop, Boston, USA, May 2008

Anja Feldmann, Vinay Aggarwal and Obi Akonjang
ISP-Aided Neighbor Selection in P2P Systems
RIPE 56, Berlin, Germany, May 2008

Vinay Aggarwal, Obi Akonjang and Anja Feldmann
Improving User and ISP Experience through ISP-aided P2P Locality
11th IEEE Global Internet (GI'08) Symposium, Phoenix, USA, April 2008

Vinay Aggarwal, Obi Akonjang, Anja Feldmann, Sebastian Mohrs & Rumen Tashev
Reflecting P2P User Behaviour Models in a Simulation Environment
16th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), Toulouse, France, February 2008

Internet Drafts

Obi Akonjang, Anja Feldmann, Stefano Previdi, Bruce Davie, and Damien Saucez
The PROXIDOR Service
Internet Engineering Task Force (IETF), Internet Draft, March 2009

Protocol Specifications

Obi Akonjang, Vinay Aggarwal, Anja Feldmann, Jun Jiang and Pengchun Xie
The Oracle Protocol
TU Berlin, September 2008

Contents

1 Introduction

Our society and life-styles are continuously being impacted in unprecedented ways by the current (knowledge-based and information/data-driven) digital age. At the forefront of this evolution is the Internet, which is the decisive and most influential technology of this era.

The Internet is a global network, made up of thousands of interconnected, but independently operated networks of various sizes. It started back in 1969 as a research project with only a few experimental nodes in one country (USA). Nearly half a century later, it has evolved into a giant multi-purpose global network with large numbers of nodes in every country on earth and is still growing in size and functions. Although its administrative structure is largely decentralized, its core role remains "centralized" by design and function. This fact is evident in its role as the current de facto medium for modern communication and information/data exchange. Its global reach and close-to-instantaneous delivery speeds, even between its furthest perimeters, makes it the most suitable and preferred medium for modern fast-paced communications. Its potentials appear to be endless.

The ability of the Internet to accommodate new technologies is a major driver of its diversity and use as a platform for growing service and application offerings. It continuously transforms and facilitates in multiple ways, the different means by which various facets of our society (e.g. businesses, government and educational institutions, political and non-political organizations, social groups and individuals) interact with each other.

1.1 Internet Growth

The Internet has been experiencing sustained yearly growths ever since its transition from a purely "research" network into an "all-purpose" (mostly business and commercial) network. With the invention of the World Wide Web (WWW) some years later, information on the Internet became easily accessible to and explorable by billions of people, boasting its popularity and growth to exponential levels. This enormous growth is portrayed in the observed rising numbers of users, connected devices, applications and services.

- **User Population**: As of August 2017, there are an estimated 3.819 billion active Internet users worldwide [186, 187], a growth of about 11.5% from the previous year. This continuous growth can partly be attributed to innovations and latest technology-trends that are fostering the development of newer, better and more user-appealing protocols, applications and services. Additional contributing factors include multiple means of connection (e.g. via fixed lines, WiFi, mobile (LTE), satellites, etc), faster speeds, and falling cost per bandwidth for end-users.

- **Connected Devices**: The number of connected devices is more than double that of users. It is estimated to reach 8.4 billion by the end of 2017 and 20.4 billion by 2020 [110]. However, in terms of mobile connections alone, active global monitoring shows the number of mobile connections and that of unique mobile subscribers to already stand at over 8.421 and 5.1 billion respectively, as of November 2017 [112]. The growing popularity and use of mobile and smart devices, together with the rapid deployment of Internet of Things (IoT), are just a few of the catalysts that are helping push these numbers to exponential levels. IoT is a technology that enables consumer/electronic devices, other than computers, smart-phones and tablets, to connect to and be controllable via the Internet.

- **Applications and Services**: The Internet's dynamic and diversified services/applications landscape avails many flexible choices, opportunities and benefits to its users and high revenues to its Providers. Popular applications and services, such as Peer-to-Peer (P2P) file-sharing, Google searches, Youtube, Facebook, Netflix and other media consumption/streaming services are contributing to observed large and ever-increasing traffic volumes [9, 108]. The Internet's growing popularity, societal importance, as well as its potential to accommodate new technologies and services, are together responsible for major shifts in both technology and business. For example, ISPs are rethinking their current business models and are re-engineering them around the Internet. Legacy infrastructures such as the circuit-switched Telecommunication networks of yesteryears, are being decommissioned and are being replaced by the Internet. Services such as telephony and radio/TV broadcasts that used to run on separate dedicated infrastructures, are now being offered as services on the Internet as well. An effect of network convergence (i.e. providing data, voice and video services on the same network infrastructure) is the massive volumes of additional voice and video traffic, which also need to be transported across access and backbone network infrastructures. When such services and applications become more popular and are embraced by even more users, their traffic portions also grow accordingly, intensifying the issues already associated with managing them.

The size of IP traffic on the Internet is currently estimated to average a colossal 122 Exabytes per month[1]. This is expected to grow to 278 Exabytes per month by the year 2021 [109]. The Internet has experienced an enormous traffic growth in the past decade, as can be seen in Figure 1.1. The global internet traffic grew by more than 22-fold, from an average of 4.234 Exabytes/month in 2006 to an average of 96.054 Exabytes/month in 2016 [99–107].

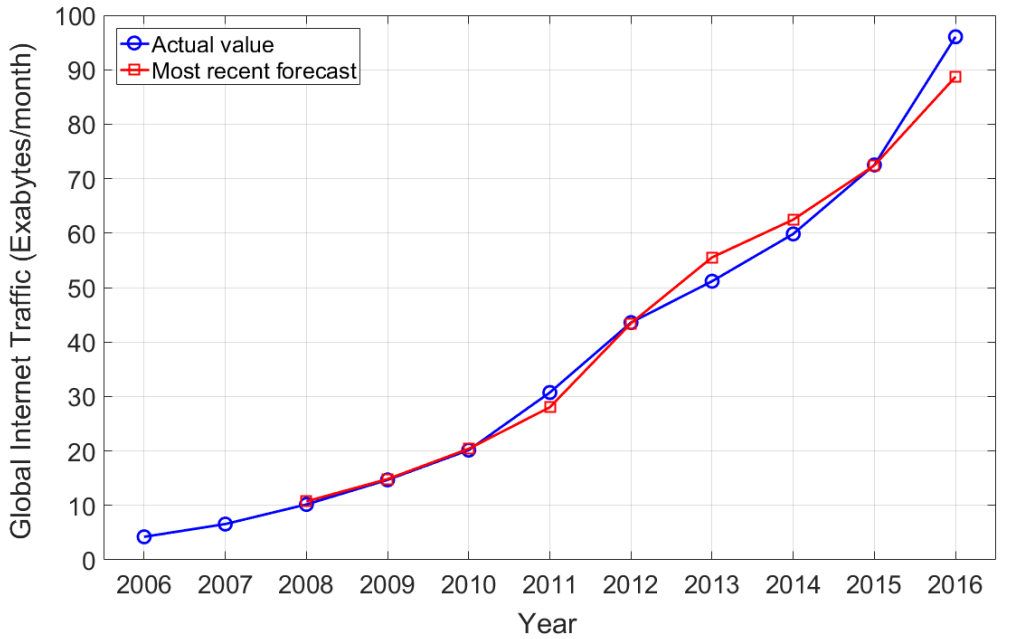


Figure 1.1: Global Internet traffic growth (Source: Cisco VNI, 2008-2017)

Year	Asia Pacific	North America	Western Europe	Central & E. Europe	Latin America	Middle East & Africa
2016	33.505	33.648	14.014	6.210	5.999	2.679
2015	24.827	24.759	11.299	5.205	4.500	1.930
Growth (%)	**34.95**	**35.90**	**24.03**	**19.31**	**33.31**	**38.81**

Table 1.1: Regional Internet traffic - 2016 year-on-year percentage growth (Source: Cisco VNI, 2016 & 2017)

In 2016, global Internet traffic (measured in Exabytes per month), grew by an approximate 32.45%, compared to the previous year [107] [109]. Regionally, the growth rates varied between 19.31% (low-end) in Central & Eastern Europe and 38.81%

[1]Based on estimates for the year 2017

(high-end) in Middle East & Africa. However, North America and Asia Pacific still lead in terms of absolute traffic volumes, as can be deducted from Table 1.1. Such observed differences in growth are sometimes a result of influential factors that are also regional in nature. For example, in North America, the region where the Netflix Streaming service was first launched, it accounted for a large percentage of all downstream Internet traffic during peak times. More than 20% in the US and a significant 13.5%, four months after it was first launched in Canada [208]. Similar trends are observed in other regions where the service becomes available as well. Netflix's traffic share continues to grow ever since, making it one of the dominant Services on the Internet today, in terms of network traffic volumes. Additional sources of sudden growth and spikes in network traffic, include special events, such as catastrophes, world sporting events and even coordinated cyber attacks. In general, long- and short-term traffic growths, as well as spikes, can cause congestions in some segments of the network, leading to delays, packet drops, jitters and other performance-related issues, if the cause is not addressed in a proper and timely manner.

1.2 Motivation

Current forecasts still predict persistent growth in traffic across all categories and regions, as can be deduced from Figure 1.2 below.

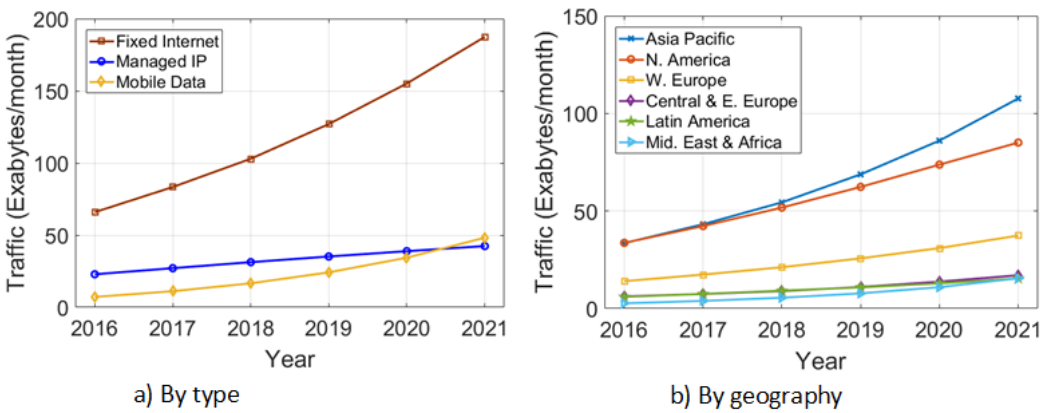


Figure 1.2: Global Internet traffic forecast (Source: Cisco VNI, 2017)

The above graphs show the global Internet traffic forecast for the coming years until 2021, classified by type (Figure 1.2a) and geographic region (Figure 1.2b).

The exponential growth in the number of Internet users and devices is accompanied by a corresponding growth in the demands for services and applications that they use. The total volume of traffic that these together generate also increases,

thereby exerting heavier loads on the infrastructures that support them. To an extent, this growing traffic load, together with the large number of applications and services, do not only impose the kind of resources they require from the underlying infrastructures, they also dictate their expected levels of performance and efficiency. However, the rate at which these supporting infrastructures are being upgraded to meet these growing demands, often lags behind that at which the requesting (access/user) devices and services are being upgraded to take advantage of faster speeds and latest technological advancements. The reasons for the slower upgrade-pace of the underlying infrastructure are many-fold and include: i) the inflexible design of these infrastructures, ii) their often very complex architectures, iii) their continuous reliance on (mostly) slow manual processes for administration and management, iv) costs.

Still and all, users (customers) on the one hand, expect the infrastructure to always be available and perform well whenever they need to use it, to remain scalable and error-free, to be secure and be able to adapt to changing conditions, irrespective of any challenges. On the other hand, rapid growth (as observed in the numbers of users/devices and the volumes of traffic flowing through the network) remains an incessant challenge to the ISP.

To ensure Service Level Agreements (SLAs) with their customers are kept and guaranteed at all times, ISPs have to pro-actively deal with the above-mentioned issues, in ways that are i) timely, ii) more efficient, and iii) cost-effective. An SLA is an agreement between an ISP and its customer, in which (amongst other things), the ISP guarantees that agreed and stated parameters, such as availability, delay, packet loss and jitters, would not exceed their contractually stated values.

Despite efforts to plan for and adapt to changing usage patterns and trends, experience has shown that network resources are never sufficient enough. This is because there are always new trends, applications or services that will potentially consume whatever bandwidth they find available. For example, the emergence of P2P, which accounts for an exceptionally large proportion of Internet backbone traffic [114] that is also quite challenging to plan for, control or manage, because of its unpredictable and disruptive nature. In addition, its high bandwidth-consumption potential often leads to bandwidth-starvation of other Internet applications. The result is performance deterioration of these applications and general dissatisfaction by other customers of the ISP. Although the global proportion of P2P traffic has reduced tremendously since some years now, it still remains the dominant peak period upstream traffic in some regions of the world [207]. In the downstream direction, real-time entertainment services are currently dominating, such as Netflix and Youtube (on a relatively large scale), as well as Amazon Video and iTunes (on a relatively smaller scale). Internet video will account for 80 to 90 percent of total IP traffic by the year 2021 [109].

1.3 Problem Statement

By reason of the above mentioned challenges, it is clear that there is a persistent need to devise faster, better and more cost-effective ways of managing the observed continuous traffic growth and their related effects. In dealing with this issue, some ISPs have even resorted to drastic measures, such as throttling, prioritizing some traffic while slowing down others, establishing pre-defined monthly limits and detecting heavy users who exceed them, in order to either limit their bandwidths or bill them [79, 217]. Such measures are rather counter-productive and turn to scare customers away. They further intensify the challenges that ISPs face, in competing with each other to win over the same group of new customers, while concurrently retaining old ones. ISPs are sometimes compelled to reverse such actions, either by self-will [36] or by regulatory/court actions against them [117]. Thus, better approaches than these are needed. Approaches that benefit both the ISP and the customer.

In this thesis, we identify the ISP backbone as a major area of interest and propose two novel approaches to help improve its performance. We use simulation studies and analyses to evaluate and demonstrate how each approach contributes to the desired goal. In the first approach, we propose the use of a simple and easy-to-implement service, the Oracle service, which effectively helps the ISP win back control of a large portion of the backbone traffic, and improve general network performance, as well as end-user experience. In the second approach, we propose a solution that exploits already available network resources to the maximum possible extent, in order to optimize traffic flows and so, improve general performance, even on very short timescales. The second approach further helps the ISP to prolong its upgrade cycles and in so doing, minimize associated upgrade-costs over longer periods, while still ensuring that SLAs and good end-user experience are maintained and guaranteed.

1.4 Traffic Management Approaches

ISPs need to plan and budget for expected, as well as unexpected growths. Infrastructural and operational changes are often involved. Changes, often first need to be planned, then tested (in a lab or a test network) and validated, before being implemented on production networks. On the one hand, such processes are quite resource-intensive and time-consuming. On the other hand, most implementations on production networks need to occur in a timely manner, i.e. before the effects of any identified issues (such as rapid growth or failure) starts to impact the network's performance. With regard to traffic growth and management challenges on backbone networks, an effective approach generally includes clever capacity planning.

Capacity planning is simply a process that ensures enough resources are provisioned and allocated to accommodate growing demands. The planning process needs to factor in all important parameters. These include close estimates of the assumed growth rates and the traffic demands that the network is expected to carry without experiencing congestion. The optimal goal remains the accommodation of all planned/unplanned growths and spikes, while limiting (or completely avoiding) congestions and failures.

1.4.1 Device Upgrades

As far as processing power and forwarding speeds are concerned, recent advances in hardware (chip and transceiver) technologies [148, 194] and software development are helping manufacturers build more powerful communication devices, e.g. backbone routers and switches, which are faster, more efficient, capable of handling larger traffic volumes and transmit at higher line speeds [150, 151, 193]. ISPs are taking advantage of the advanced features and capabilities of these devices to redesign their architectures, re-engineer their infrastructures and simplify/automate administration and management. With these new devices, they aim to gain more flexibility, higher efficiencies and better overall performance. These advanced features and innovative designs have, for example, led to recent shifts towards Software-Defined Networking (SDN), virtualization and cloud-hosted (instead of inhouse-hosted) services.

1.4.2 Bandwidth Provisioning

The capacity of the network is a measure of the maximum amount of data that could be transported between locations on the network. For backbone networks, the most important resource is the link bandwidth. This translates into ensuring that bandwidth is sufficiently (over-)provisioned across all backbone links.

A simple approach often used by ISPs, is collecting utilization statistics of core links and upgrading them based on a simple rule of thumb principle, such as when their average utilization attains 50% or some other ISP-determined target. However, with this approach, the ISP is not optimizing on their investments, as more capacity is often provisioned than is really necessary. Additionally, there are still no guarantees that links which are already over-provisioned using this approach, are also provisioned enough to deal with link and device failures [166, 192].

A better approach uses methodologies that determine bandwidth requirements to meet SLA goals, while also taking influential parameters, such as link and device failures into consideration. The goal here, is to maintain performance and scalability at all times, even during failures, while concurrently minimizing the capacity that

has to be over-provisioned. This further satisfies another important goal, which is to minimize the overall cost associated with over-provisioning.

1.4.3 Adding New Links

Another simple approach sometimes used by ISPs, is the addition of new links to the topology. This can be done by ether adding to already connected nodes, i.e., creating parallel links or by creating new links where none existed before.

- **Parallel Links**: Parallel links refers to new links that are added between locations that are already directly connected with each other. This is often required when the current link(s) have maxed out their available capacity, such that bandwidth provisioning can no longer be performed on them. Addition of new link(s) to form parallel links, becomes a feasible alternative to increase the bandwidth between the two locations. Although this option has little to no architectural impact on the existing topology, it still influences the routing protocol, by enabling it take advantage of the added link(s) (bandwidth) to re-adjust its routing metric, which in turn affects the traffic flow.

- **Non-Parallel Links**: ISPs can also create completely new links between locations that are not yet directly connected with each other. Such are referred to, as non-parallel links. Since this option brings changes to both the network architecture and the topology, the ISP needs to first analyze its impacts on the network as a whole before implementing. Prior planning is thus necessary, which also adds to its complexity. Ignoring it could lead to unwanted effects.

Both of these mentioned options are usually preceded by mid to longterm planning and associated with costs, which often also need to be justified and approved. A short timescale solution using this method is thus quite unlikely.

1.4.4 Flow Rerouting

Flow rerouting is the process of changing the paths that flows take, either as a response to changes in network conditions or as a means of achieving a desired Quality of Service (QoS) goal. A typical scenario involves traffic demands with flows between domains, i.e. flows that transit via dedicated ingress links through to a set of egress links on the ISP's backbone. The demand model allows prediction of how changing the internal routing impacts the distribution of load on the ISP's backbone links [53]. Flow rerouting is also used to avoid bottlenecks. The advantage of this approach is that it could be used on a smaller (shorter) timescale.

1.4.5 Change of Traffic Matrix

The Traffic Matrix (TM) of a communication network is a measure of the total amount of traffic between all possible Origin and Destination (OD) pairs (or nodes) of the network. It is an important input component for optimal network design, capacity planning and traffic engineering [152, 205]. An accurately measured TM is an important and critical tool, used by ISPs to predict future traffic trends, detect anomalies and perform network optimization.

A change of Traffic matrix, as a result of changing where traffic enters and/or leaves a particular domain, is another approach used to manage/control the flow of traffic across an ISP backbone. Although this is a technically feasible solution, contractual agreements and already implemented routing polices might need to be verified and adjusted first before implementation, which is a potential hindering factor to a timely implementation.

1.4.6 Flow Optimization

An IP traffic flow is a sequence of packets of common source that at any given time, are passing through a common path (or link(s)) to arrive at a common destination. It can generally be identified by means of a 5- to 7-tuple, which includes the source address, the destination address, the source port, the destination port, the layer 3 protocol, the class of service and the device (router or switch) interface, with all but the last, being attributes of the IP packets.

Flow optimization uses a variety/combination of approaches to control the flow of packets on the network.

Optimization using The Oracle Service

The emergence, rapid growth and disruptive nature of P2P traffic on backbone networks, coupled with their ability to establish overlay networks that are completely agnostic of the underlay network [5], posed huge management as well as capacity-planning challenges to Providers. The traffic overhead of P2P systems is relatively high. One reason for this, is the attempt by peers to infer network condition themselves, as a means to improve performance. The information that P2P nodes need, but can't accurately infer, is the same information that ISPs already possess, but won't publicly share.

In order to limit the disruptive nature of P2P traffic and curb their negative impact on backbone networks, a proposal to enable collaboration between P2P systems and the ISP is made [4]. An approach based on this proposal is the **Oracle service** [8]. It is a proximity service hosted by the ISP and freely offered to P2P nodes, to aid them locate and select 'better placed' neighbors on the overlay networks. The

Oracle service acts as the collaborator that 'passes' ISP information to the peers in an unconventional but secure manner. By expressing a preference with respect to locality, it helps P2P nodes make better decisions, when selecting potential neighbors or sources to download content from. It does this by sorting out and expressing a preference based on locality, using information originally supplied by the requesting peer. It does not directly send network-related data to the peer and thus prevents ISP information from being compromised. Confidentiality is therefore maintained. Peers that use the Oracle service benefit from the ISP's knowledge of the underlay network to establish more coherent overlay networks, eventually leading to network performance improvements and better user experience. The ISP also benefits by gaining increased influence and control over this huge "disruptive" constituent of traffic flowing via its (backbone) network [3]. Regaining control of a huge proportion of its traffic increases the ISP's ability to more effectively engineer it, so as to retain most of the traffic within its own AS domain and save on transit costs (if/where applicable). The ISP can now also plan better and offer better services to its other customers.

We note here that a similar approach to the Oracle service, named **Provider Portal for (P2P) Application (P4P)**, is proposed in [214]. They propose a collaboration platform, in which *iTrackers*, owned by individual ISPs and *appTrackers* in P2P systems, communicate and share information to improve the performance on both sides. ISPs feed their iTrackers with network-related information that P2P clients can retrieve through querying using their appTracker.

There are fundamental differences between the Oracle and P4P approaches, in the method and implementation details of the collaboration. In P4P, the ISP collaborates with the P2P user by passing on network-related information (secrets) to the peer. We argue that this poses potential risks to the ISP. Since, giving out such private information, could in extreme cases, be exploited and used against the ISP. Our approach with the Oracle service, offers the same service, but with the added advantage of not needing to reveal any ISP-related network secrets to the peers.

Optimization using Provider-aided Distance Information System (PaDIS)

As usage patterns shift from P2P file-sharing to media consumption, Content Distribution Infrastructures (CDIs), which handle media distribution to end-users, are increasingly being challenged as well. CDIs have to optimize their operations to accommodate growing demands, while still guaranteeing optimal user experience. Poese et al [162] based their work on the same approach as the Oracle, to enable ISP/CDI collaboration, with the ISP offering a similar kind of service to CDIs. The new service, named **Provider-aided Distance Information System (PaDIS)**, is hosted by the ISP and allows collaborating CDIs to obtain needed mapping and other operational information related to the ISPs' infrastructure, without the ISPs having to reveal any of their operational secrets.

Application Layer Traffic Optimization (ALTO)

The ALTO working group was created by the Internet Engineering Task Force (IETF). Its goal is to merge the different optimization proposals and work out a common standardized ALTO protocol. So far, the ALTO protocol [10] and deployment considerations [188] have been proposed in RFC7825 and RFC7971 respectively.

Optimization Using Mixed-Integer Programming (MIP)

The often observed traffic upsurges and momentary spikes on backbone links, continue to pressure ISPs for shorter timescale solutions that are also cost-effective. With Mixed Integer Programming, optimized metrics for efficient routing of traffic flows and distribution of load on the network, could be determined in a matter of minutes. The advantages of this approach include; its speed, exploitation of already available resources and the fact that no time-consuming and expensive physical topological changes are involved. More information on this approach is provided in chapter 6.

1.5 Summary of Contributions

Studies that investigate correlations between the overlay networks formed by P2P systems and the underlay networks of the ISP, show that neighbor-relationships in the overlay network are either randomly [5] or at most selfishly [180] formed. This is in stark contrast to how they are formed in the underlay network. Further analyses reveal that a large portion of these neighbor-relationships are formed between peers that belong to different ASes, although other potential peers exist in their same AS and even in their same location.

Based on these foundational works and other reported findings, we propose a solution that fosters cooperation between P2P users and ISPs, as well as improve the correlation between the overlay and the underlay networks.

With shifting demand patterns and new trends so far warranting different streamlined approaches to tackle, what is fundamentally an old challenge, we propose another approach that offers a shorter timescale solution and makes use of existing resources to optimize the flow of traffic on backbone networks. It employs Mixed-Integer Programming to determine best routing costs for optimized traffic flow.

The following summarizes the main contributions of this thesis:

The Oracle Service

We propose a new and freely offered service, the Oracle service, which is hosted by the ISP and helps peers make informed decisions, as to which peers they preferentially should connect with, when joining the overlay or which peer to download from, after a search with results from multiple peers. The ISP knows its network best and through the Oracle service, can freely offer this information to peers, in a form that does not reveal internal details. That is, details that could render the network vulnerable to attacks or cause a business disadvantage. The peers benefit from this service because it saves them the need to infer the same network properties themselves [47, 172], a process which is often much tedious, but less accurate. Localization by preference is one of the services offered by the Oracle. Peers are able to preferentially select neighbors that are in the same AS and the same (or nearest possible) location like themselves, based on informed decisions made possible by the Oracle service. In effect, the service helps localize the traffic between peers and through that, enable the ISP to retain a good portion of the overlay traffic within its own domain. This is a win-win situation, since localized traffic improves download response times for the peers, while also reducing transit costs for the ISP.

Analyses of Peer-to-Peer/ISP Collaboration

We conduct packet-level simulations to analyze the proposal and quantify the performance improvements for both the P2P user and the ISP.

Comparative study of traffic effects on different backbone topologies

We investigate the effects that huge traffic flows generally have on different backbone topologies. Using a national backbone network model for reference, we study how three derived topologies are affected by the same volume of traffic. All three topologies have 12 nodes, but differ in the number of their links and how these links are connected. A fullmesh topology with 66 links and two partial-mesh topologies with 30 and 20 links respectively, are created. Our analyses show that the fullmesh topology with 66 links is an overkill, as it performs best, yet in many cases, its performance remains comparable to those of the partial-mesh topology with only 30 links. With increased traffic, the topology with 20 links (the least number of links) is observed to be the one that is also most affected by congestion, leading to performance degradations, a phenomenon which is not (or only minimally) observed in the other two topologies.

Flow optimization using Mixed-Integer Programming (MIP)

Understandably, the topology that has the least number of links (i.e. 20 links) and the least amount of total bandwidth of all three topologies, is also the one expected to offer the least performance. However, further analysis of this topology reveals that while some links are suffering from over-utilization, others carry little-to-no traffic at all. We thus investigate if comparative gains could still be attained if the traffic flow is engineered differently. We therefore propose a method that employs Mixed-Integer Programming to help determine and select optimal flow paths through the network. We carry out further simulation studies to assess and compare its effects. Our findings show that comparative gains are attainable using this method.

1.6 Structural Overview

The rest of the thesis is stuctured as follows:

Chapter 2 provides background information upon which the thesis is based. It also presents Internet trends that are currently impacting both end-users and Service Providers.

Chapter 3 presents structural properties of backbone networks and the challenges they face with ever-growing traffic. It also presents the solutions/discussions that are helping to tackle these issues.

Chapter 4 specifically investigates the case of Peer-to-Peer as a major backbone traffic contributor. It describes the Oracle service and shows how it functions as an enabler for Peer-to-Peer and ISP cooperation. It then presents the simulations done in support of this concept and provides their results and analyses.

Chapter 5 investigates the performance of different backbone topologies under the same traffic conditions. In order to study these effects, the different topologies and traffic conditions are modeled using a popular network simulation tool. It then presents the results obtained from studying the effects of increased traffic and single link failure on each of the topologies.

Chapter 6 looks deeper into the least performing topology of Chapter 5. It presents our proposal of using Mixed-Integer Programming to optimize traffic flow and improve the performance in this topology as well. The simulation results and analyses that support the feasibility of the proposal are also presented.

Chapter 7 summarizes our results and the conclusions drawn from our findings. It also discusses and provides directions for future research.

2

Overview of the Internet

In this chapter, we present a general overview of the Internet, including basic concepts and background information that are relevant to this thesis. We start with its structure and a general classification of its entities (Tiers) and owners (Internet Service Providers). We next look into the kinds of relationships/interconnections that ISPs have with each other, the routing protocols they use for communication, including the standard TCP/IP protocol suite used by systems on the Internet. The different types of Service Providers are also presented, followed by the different types of services and applications that they offer to their customers. Some of these applications/services are fundamental to the Internet's operation, while others are quite popular with end-users. We conclude the chapter by outlining some of the common metrics that are used to assess performance on the Internet and elaborate on how they are measured.

The Internet is a global network of interconnected autonomous networks (or autonomous systems). An *Autonomous System (AS)* is a group of networks under the same administrative control. These often also share the same external routing policy. Most ASes are owned and independently operated by Internet Service Providers (ISP) for profit. There are currently over 60 thousand registered Autonomous Systems on the Internet today [33]. The Internet's size (and continuous growth) has been and continues to be a topic of main interest to both researchers and operators.

2.1 Internet Structure

Since its inception, the Internet has evolved into a roughly hierarchical, but yet a complex structure of interconnected networks. Agreements between ISPs, including the policies that they make and implement, partly account for the Internet's structural architecture, as well as the direction and speed of its evolution.

The various ISPs that operate the Internet can be classified in many different ways. At the AS-level, they can be classified into one of three major hierarchical *tiers*,

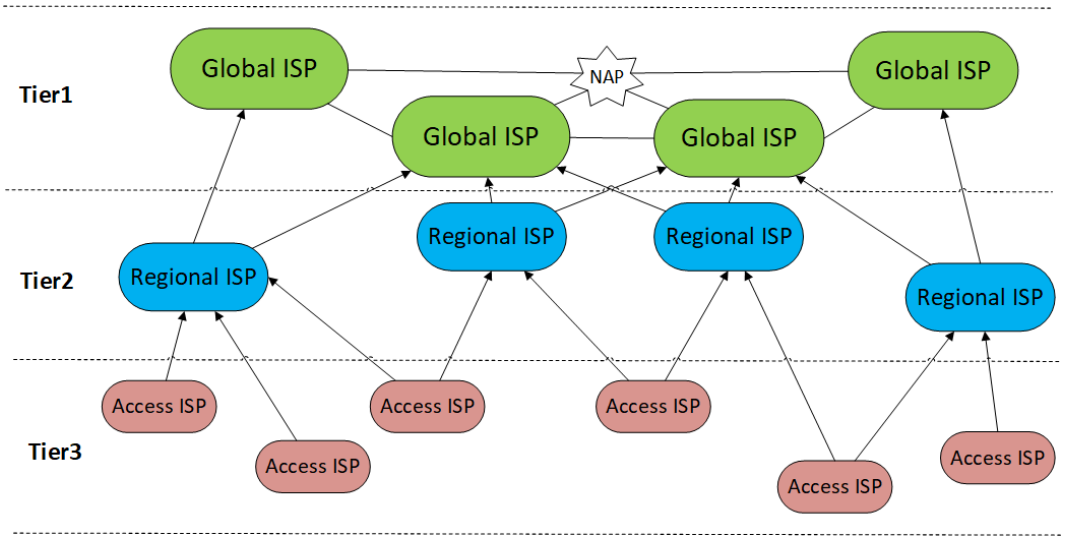


Figure 2.1: Traditional Internet Structure

depending on their role, the size of their AS and their geographical footprint. These hierarchies range from Tier1 (top of hierarchy), over Tier2 (middle of hierarchy) to Tier3 (bottom of hierarchy).

Tier1 (or Global) ISP: Tier1 ISPs manage very large networks that spread across multiple continents and large geographical areas. Only a small group of ISPs fall within this category. Each Tier1 ISP connects directly to all the other Tier1 ISPs as equal partners (settlement-free).

Tier2 (or Regional) ISP: Tier2 ISPs are regional in scope, i.e. they operate within a defined geography, which is less than global. Their geographies are usually national or continental, but not global, as it is with Tier1 ISPs.

Tier3 (or Access[1]) ISP: Tier3 ISPs operate in the last-mile. They operate at the edge of the Internet and provide access to businesses and homes. Their scope of operation is geographically limited to towns/cities, provinces or national boundaries.

Interconnections between ISPs are mainly driven by economic incentives. As a result, the Internet's structure is also evolving. The traditional structure in Figure 2.1 is evolving into the recent more flatter (traffic-driven) structure shown in Figure 2.2. The Network Access Points (NAPs), which were public facilities where ISPs connect with each other for peering, have long been replaced by current-day Internet Exchange Points (IXPs). ISPs generally establish one of two major kinds of

[1]Access ISP increasingly refers to the role than to the type of ISP, since some Global ISPs and most Regional ISPs also often have internal business units that offer the same line of services.

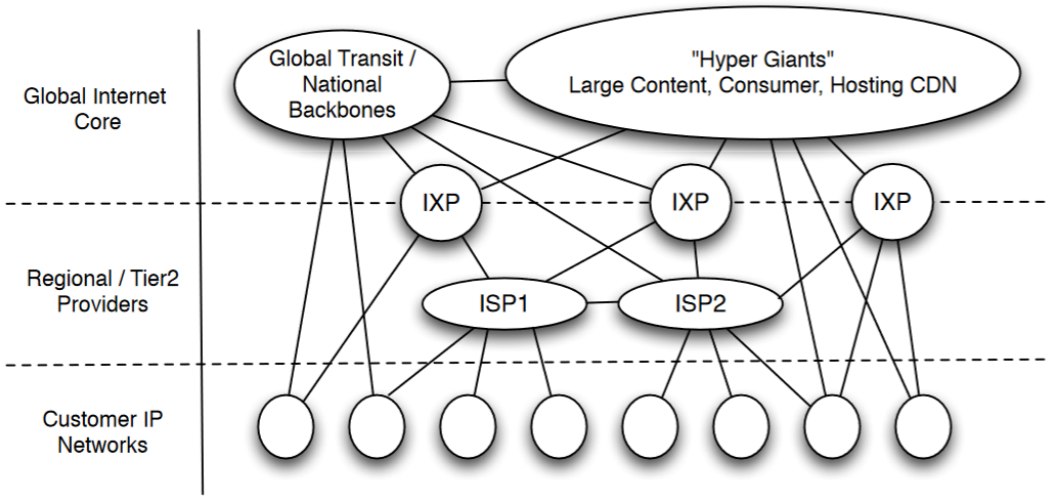


Figure 2.2: Recent Internet Structure - illustrating dominant Internet traffic patterns [130]

interconnection relationship with each other, i.e. either a 'transit' or a 'peering' relationship.

A ***transit*** relationship involves the payment of a settlement fee for transport services. It is usually established between ISPs of different tiers, such as between a higher-level Tier1 ISP and a lower-level Tier2 ISP or between a Tier2 ISP and a Tier3 ISP. The lower-level Tier2 or Tier3 ISP pays the higher-level Tier1 or Tier2 ISP, respectively, to carry its traffic to the rest of the Internet. The amount of fees paid is usually proportional to the volume of transit traffic that is transported, i.e., the higher the volume of traffic, the higher the fees.

In a ***peering*** relationship, two ISPs agree to a settlement-free exchange of routing information and traffic. They both share the cost for the connection(s), but none pays the other for the volume of traffic exchanged. This kind of relationship is common between ISPs of the same category, e.g between Tier1 ISPs or between Tier2 ISPs in the same (or neighboring) regions. The main driver behind this kind of relationship, especially between non-Tier1 ISPs, is to avoid or minimize the cost for transit.

Generally, the normal practice is such that Tier2 ISPs in the middle of the hierarchy, purchase transit services from Tier1 ISPs and also offer transit services to even lower Tier3 ISPs. It is also becoming more common for Tier3 ISPs to purchase transit services directly from Tier1 ISPs. To manage transit costs, both Tier2 and Tier3 ISPs respectively also enter into peering relationships with other ISPs of the same tier. It should also be noted here that, just like Tier3 ISPs, some Tier1 and Tier2 ISPs also offer access services to businesses and homes.

To interconnect, each ISP first needs to register its AS with the appropriate Regional Internet Registry (RIR). It then obtains a public Autonomous System Number (ASN) and can then use it to enable communication and routing information exchange between itself and its peering/transit partners. Thus, while each ISP can independently decide on how routing within its own AS should occur, in oder to interconnect with other ASes and provide a global reach to its customers, it is compelled to adhere to standardized guidelines that stipulate the use of ASN and an Exterior Gateway Protocol (EGP) to enable an interconnection. Irrespective of the AS that remote systems and users belong to, end-to-end communication and data exchange between them is made possible through this means.

2.1.1 Edge and Backbone Networks

Networks, such as the Internet, have two major (physical) elements of great significance: i) the links through which data flows and ii) the switches/routers that control the flow of data on these links. Many different kinds of media can be used to establish these links, ranging from wireless (such as Radio Frequencies (RFs) in LANs and satellites across larger geographies) to wired (such as copper, coax and Fiber cables for LANs and WANs). The type of media used, usually depends on the requirements for that segment of the network. At the edge, i.e. at the entry point of the network, access to many different customers is needed, thus the quantity and variety of available entry points is comparatively more important. In the core of the network, the capacity to handle aggregated traffic from the edge, is more important. Thus, speed and capacity are more important.

Taking these into consideration, the Internet can also be classified into edge versus core networks. That is, a customer-facing edge, consisting of a large number and variety of access links of small-sized to medium-sized bandwidth and a provider-managed core, consisting of a much smaller number of aggregated high-speed links of much larger bandwidths/capacities.

- **Edge Networks**: An edge network is a network that is located on the periphery of an ISP's network. It demarcates the entry point for traffic flowing from customers and peers networks, to/through the ISP's own network. Edge networks can generally be classified in 2 main categories; *access* network that carries traffic from/to home and business customers and peering interconnections that carry traffic from/to other ISPs.
- **Backbone Networks**: Backbone networks carry the bulk of all the traffic that traverses the Internet. They are characterized by very large capacity high-speed links and high-end backbone routers that can handle large numbers of aggregated flows of different classes. Their main design and operational goals include; high availability, scalability and resiliency when faced with link and/or device failures. Backbone networks are therefore expected to possess

multiple paths between any two PoPs, in order to accommodate and mitigate such failures. That is, failures should not negatively impacting performance, e.g through additional delays or packet loss. Since distances between interconnected PoPs could be quite large, re-routing during failure, could mean taking an even longer path, which turns to increase the delay that packets experience. Rerouting is thus done as a last resort. A better approach that is commonly used to avoid this kind of issue, is to run multiple (and often disjoint) links between two adjacent PoPs, so that in case of a link failure, the traffic load could evenly be shared among the remaining operating links.

2.1.2 Internet Exchange Points

An Internet Exchange Point (IXP) is a location where different autonomous networks physically interconnect with each other to exchange traffic. They contribute a lot to the structure of the Internet and play a significant/facilitating role in peering between ISPs, CDNs and Providers of other Internet services [24]. Ground-breaking and insightful analyses of their traffic is offered in [2]. IXPs are usually dispersed across a country or region. This creates proximal exchange points for local traffic and thus eliminate the need to exchange local traffic in further away or even oversea locations. IXPs also create a unique location to host other essential Internet services, such as DNS, Web caches, time servers, root server mirrors, etc, because of the proximity to the connected networks and users. As a result, IXPs enable faster switching/routing speeds, faster access to content and hosted services, better user experience and cost reductions for the participating ISPs. They also provide the appropriate location to install vantage points in the Internet [25].

2.2 Service Providers

The Internet is basically a kind of service marketplace, where service providers and service consumers come together to respectively sell and buy service-commodities. The nature of such commodity could be commercial, educational, for business, for pleasure or many other options. A fundamental requirement for the consumer, which is also the most common service offered by an ISP, is universal access to the whole Internet via the local ISP's own autonomous network.

Despite global coverage by some very large ISPs, there is none that can provide universal end-to-end access without collaborating with at least one or more of the other ISPs. These collaborations are strategically business in nature and result in contractual agreements that state the conditions and price for exchanging traffic between their ASes.

A Service Provider (SP) is a company that offers specific services to businesses, organizations or individuals on the Internet for a fee. Service Providers are often

classified by the type of service(s) that they offer. These include; Internet access, hosting, content distribution, cloud services, video on demand and many others. Some SPs offer a combination of these services, while others offer only a single type and are classified by that one type alone. For example, a company that offers Internet access services, is generally referred to as an **Internet Service Provider** (**ISP**). However, big ISPs often offer one or more of the other services as well.

Thus, based on the specific service offering, Service Providers can also be categorized as follows:

- **Network Service Provider (NSP)**: NSPs are businesses that offer packet-forwarding services and Internet Protocol (IP) services on the Internet. They include *access providers*, who provide internet access services to businesses and individuals and *backbone providers*, who operate large global networks and provide transit services to other (smaller) providers, even across long distances. NSPs are responsible for creating and managing Internet connectivities.
- **Application Service Provider (ASP)**: An ASP offers upper layer applications or softwares, such as email, instant messaging or web-based training, which require Internet access as a primary condition for use.
- **Hosting Service Provider (HSP)**: HSPs offer web-hosing services over the Internet.
- **Content Distribution Service Provider (CDSP)**: A CDSP is a provider who offers speedy delivery/distribution of web and rich media contents to end users. They build Content Delivery Networks (CDNs), which are overlay networks operating on top of the IP underlay network, allowing them to distribute content with no need to manage the underlay themselves.
- **Content/Information Provider**: They are owners of the content or information that is distributed/transported by the CDSP or the NSP respectively, to the end users. The types of content or information range from web portals to video/audio data and also services such as Google search and wikipedia.
- **Cloud Service Provider (CSP)**: Cloud Service Providers are businesses that offer network, infrastructure or application/software services in the cloud. These services are accessible to subscribed customers only via the network.

Obviously, no single Provider can alone offer all the services that all customers need. However, through network interconnections, service collaborations and business partnerships, a large number of these services can be transparently offered to requesting customers, while the details of any involved collaborations are kept private. Since the system offering a particular service and the consumers of that service are often not in the same location or belong to the same ISP, the network plays the fundamental role of being the medium, through which communications and information exchange between these systems happen. End-systems on the Internet that

wish to communicate with other end systems, need to use standardized protocols. TCP/IP is the suite of protocols that has been approved and standardized for this purpose.

2.3 The TCP/IP Protocol Suite

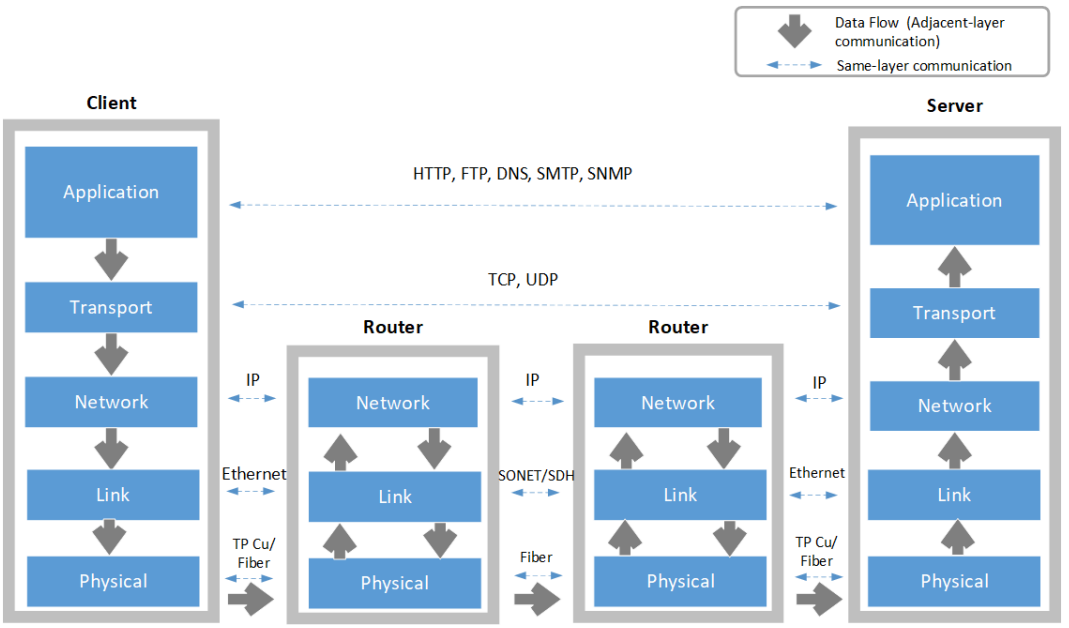


Figure 2.3: TCP-IP Protocol Suite with client-Server interaction

The Internet generally operates via implementation of standardized protocols. A protocol is a set of syntactic and semantic rules or procedures that govern how communications occur. Computers and all other devices on the Internet, use such protocols to communicate and exchange data with each other.

The Transmission Control Protocol (TCP) and the Internet Protocol (IP) are two outstanding protocols, among others that belong to the suite of protocols, collectively known as the **TCP/IP protocol suite** [129]. Figure 2.3 illustrates the TCP/IP protocol suite and its use in a typical communication between a client and a server.

The TCP/IP protocol suite consists of five layers[2], which together provide interconnection and communication services between devices [65] [19] [51]. They define the

[2]TCP/IP is traditionally represented by a 4-layer model, i.e. Application, Transport, Internet and Network Interface layers. However, modern literature increasingly uses an updated 5-layer model, with the *Network Interface* Layer being split into the *Link* and *Physical* layers

syntax and formats of messages, e.g. requests and responses, that are exchanged between devices and also specify how these handle errors, should they occur [51]. The main purpose of TCP/IP is to enable devices, which are usually different by vendor, hardware, architecture and purpose, to communicate with each other, despite these differences.

Application Layer: This is the topmost layer and the layer at which an application program interacts with the network to send and receive data. Application layer protocols address the formatting of applications and the commands/responses that the systems offering them must support.

Transport Layer: The Transport layer provides transport services to the Application layer. It ensures that data passed down to it from the Application layer is transfered appropriately to the intended destination and vice versa. Two common protocols used to accomplish these tasks are the **Transmission Control Protocol** (TCP) and the **User Datagram Protocol** (UDP). TCP is connection-oriented and is used to provide reliable transfer between source and destination end-systems. On the other hand, UDP is connectionless and is used to provide simple (and unreliable) data transfer between end-systems. Two additional transport-layer protocols that are relatively newer, but less used, are the **Datagram Congestion Control Protocol** (DCCP), specified in RFC 4340 [126] and the **Stream Control Transmission Protocol** (SCTP), specified in RFC 4960 [204], respectively.

Network Layer: This layer provides addressing and routing services to the upper Transport layer. The main and only significant protocol at this layer is the Internet Protocol (IP). Segments received from the Transport layer are encapsulated into IP packets, each with a header that contains both source and destination addresses, as well as other important information. Intermediate systems such as routers, use these information to determine how to forward the packet to its next hop. This is repeated on a hop-by-hop basis, until the packet's final destination. There are two versions of the IP protocol, the older 32-bit **IP version 4** (IPv4) and the newer 128-bit **IP version 6** (IPv6). IPv6 was created to address some short-comings of IPv4, such as address-shortages and security.

Link Layer: The link layer provides physical addressing, framing and error detection services. It encapsulates IP packets into frames with hardware (link) addresses that enable forwarding across the local link.

Physical Layer: This is the lowest layer and that at which the physical transmission medium exists. Frames from the Link layer are converted into raw bits and then transmitted over a communication channel on a local medium. Typical media include Twisted Pair Copper (TP Cu) and Fiber cables.

respectively, and the *Internet* layer renamed to the *Network* layer. Overall, both models contain the exact same functions

2.4 Internet Applications and Services

In the early days of the Internet, the service spectrum was quite meager. It consisted only of a few services, i.e. remote login, remote file access and electronic mail. However, with time and technological advancements, many others have been added, a good number of which are very popular and in widespread use today.

Most end-users perceive the Internet primarily through the set of applications and services it offers them. Although some of the implemented protocols are quite complex, the use of Graphical User Interfaces (GUI) by most applications successfully hides their details and complexities from the end-users. Therefore, to effectively use their services, end-users neither need to understand the details of their protocols nor know how they use the underlying network for successful end-to-end communication.

Network applications generally use two kinds of interaction models; the Client-Server model and the Peer-to-Peer model.

2.4.1 Client-Server Application Model

In a Client-Server application model, the tasks performed by the service as a whole, are divided between a *server* and a *client*, with the server providing some type of service to the client.

A *server* is generally an application program that offers a particular set of services over the network to requesting clients. It waits for requests at a well known port reserved for the service, processes them when they arrive, sends back the response to the client, then go back to wait for the next request. The speed at which these happen, defines the efficiency and performance of the servicing process.

A *client* is an application program that requests for and makes use of the services offered by the server. A client that needs to use a service, e.g. download a file, first composes a request, sends it to the server and then waits for the response from the server. Figure 2.3 illustrates how such a request is created by a client and forwarded across the network onto a server. The request is sent to the server via a well-known (reserved) port. The server accepts the request (if it is valid), creates a response and sends it to the client.

2.4.2 Peer-to-Peer (P2P) Application Model

Network applications that are based on the P2P model, are designed to be more distributed in their functioning. There are no single servers that solely operate as such. The functions of the server (and of the client) are distributed among all participating peers, i.e. each peer can simultaneously act as both a client and

a server. A peer can thus send its request for a service to other peers and can concurrently also respond to requests coming from other peers. If a peer receives a request that it can't directly respond to, it simply forwards it onto others (neighbors) that likely can. This therefore eliminates the need for a central server.

The P2P model shares some similarities with the *client-server model* described in section 2.4.1 above, since its tasks are also distributed between clients and servers, just like in the traditional sense. However, they also have major differences, in that, the functions of the server are no longer on a dedicated centralized system, but are replicated on all active peers of the overlay network.

P2P applications are typically used for file-sharing, multimedia streaming, telephony and gaming.

2.4.3 Popular Application Services

Common services with applications based on either the client-server model or P2P model include:

Domian Name System (DNS): DNS [143] is an Internet naming service, used to convert IP addresses into easily recallable high-level names and vice versa. Machines on the Internet are normally identified by their unique IP address. Human users would therefore need to memorize millions of IP addresses to be able to connect to them, which is quite a difficult thing to do. To remove this difficulty, DNS servers are used to convert these IP addresses into names that humans can easily memorize and recall.

World Wide Web (WWW): WWW (or simply 'The Web') [16] is the most popular and most widely used service on the Internet. In fact, most users take it to be the Internet itself, although it is only one of many services that the Internet offers.

The web is a collection of distributed resources (documents and services) that are scattered across the Internet, but linked together via hypertext links. Its three main components are the web server, the web client and the transfer protocol that both the server and client use for interaction.

- The **web server** is where all the resources are stored. The Universal Resource Identifier (URI) is a string of characters that are used to identify these resources. The Uniform Resource Locator (URL) is a unique name (or web address) that is used to specify the exact location of a resource on the network.
- The **web client** is an application installed on a local system and used to access, download and display resources that are stored on the web server. Browsers such as Firefox, Google Chrome, Internet Explorer, Safari and many others are examples of web clients.

- The **Hypertext Transfer Protocol (HTTP)** [57] is an application-level protocol for distributed, collaborative, hypermedia information systems [56]. Web servers and clients use it to communicate with each other.

Taking the URL *'http://www.example.com/example.html'* as an example, a web client such as the Firefox browser, uses the application transfer protocol 'HTTP' to establish a connection with the web server named 'www.example.com', in order to access, download and display the URI 'example.html' and all its embedded objects on the local machine.

Electronic Mail (email): Email is a popular Internet-based method of exchanging messages (mails) via the use of electronic devices, such as computers, tablets and smartphones. The delivery service is based on the Simple Mail Transport Protocol (SMTP) [122]. Emailing is very fast, with delivery occurring within seconds and without the need of human intervention. The normal (postal) mail delivery system requires direct human involvement and depending on the distance/location, could require a couple of days for the delivery to be completed. The postal method of mail delivery is also known as "snail mail", referring to its slow delivery speed, compared to that of emails. Initially, email allowed users to exchange only short text messages, but with time, it has evolved to allow much longer messages, including embedded objects, such as images and sound.

File Transfer: File transfer involves the copying of computer files from one machine onto another. This service is also based on the client/server model. The client sends a request to the server, asking for the file and starts downloading a copy, if the server acknowledges. An Internet standard-based protocol, the File Transfer Protocol (FTP) [164] is used to deliver the file from one end-system onto another. Other popular protocols that are used to transfer files include; FTP Secure [64, 88], Trivial File Transfer Protocol (TFTP) [181], HTTP, HTTPS [170, 171] and Secure copy (SCP) [200], which is based on Secure Shell (SSH) [215].

Remote Login: The remote login service offers the ability for a user to log into a computer system, as an authorized user, while not being physically present in the same location as the system. The three most popular protocols for remote login are rlogin [118], telnet [165] and Secure Shell (SSH). With rlogin and telnet, all exchanges between the local and remote systems are sent in clear text. These are less secure as it poses the risk of the communication being eavesdropped without detection. Meanwhile, SSH overcomes these shortcomings by providing options for strong authentication and strong encryption that protect the communication's security and integrity.

File Sharing: File sharing is the process of offering access to digitally stored resources using an appropriate distribution protocol. The stored resource could be documents, ebooks, computer programs, graphics, as well as audio and video files. On the Internet, this distribution could occur either via download through a hyperlink, download from a file hosting server or using a file sharing P2P application. A

downside of file sharing is the imminent risk of being infected by viruses, adwares and spywares. These often get installed on the downloading computer without the user knowing.

Media Streaming Services: Media streaming involves the transmission of multimedia data (audio or video), from a source (server) system onto a destination system (or player) for immediate consumption. The original data is compressed and sent in continuous streams that get played immediately upon arrival. The user does not have to first save the data before playing it.

Online Gaming Services: Online gaming refers to the act of playing video or role-playing games either partially or fully via the Internet or another network. Online games can be classified into different categories. **Browser games** are simple games that can be played using a web browser. With the support of web-based graphic enhancers, such as Java and Flash, more complex games are also being developed for the browser. **Real Time Strategy (RTS) games**, such as *Starcraft* and *Age of Empires*, have native Internet support that enables connected players from all over the world to play with/against each other.

Cloud Storage: Cloud storage is a cloud computing model in which data is stored on remote servers accessed from the internet or "cloud". It is operated, managed and maintained by a cloud storage service provider. The service runs on storage servers that are built using virtualization techniques.

2.5 Packet Forwarding

In the history of the Internet, competitive methods have been developed for the effective forwarding of IP packets. Each method differs from the other in one way or another. Newer methods are developed to overcome drawbacks of already existing methods. Older methods are upgraded to address newer requirements, such as the introduction of IPv6. The main goal of the various methods is generally to address existing issues in particular scenarios or simply to offer alternatives that employ better metrics for improved convergence, better scalability and better performance.

Generally, IP forwarding can be done by means of routing or more recently by means of label switching.

2.5.1 IP Routing

IP routing refers to the process of forwarding IP packets along a determined path from their respective sources to their intended destinations. On the Internet, dedicated devices (routers) that communicate with each other to exchange connectivity and link quality information, are used as the primary packet forwarders. Routers use routing protocols to advertise their local subnets to their neighbors and receive

the same about remote subnets from their neighbors. If more than one route exists to a particular subnet, the best one is selected through the routing protocol's routing algorithms, then added to its routing table for use when forwarding packets. In case of changes in the topology, e.g. due to link or device failure, by which active routes become unavailable, the routers react by advertising these to their neighbors and then select the next best route as a replacement. After such a change occurs, the time it takes for all routers in the network to settle down with the next best route is referred to as the convergence time. The quickness with which a routing protocol converges is an important criteria when selecting potential routing protocols for backbone networks. Other important criteria include support for summarization and the ability to scale properly in very large environments.

Static versus Dynamic Routing

Routing tables can generally be populated using three distinct methods; (i) directly connected routes that are automatically added, (ii) static routes that are manually added and (iii) dynamic routes learned and automatically added by dynamic routing protocols.

Administrators who want to determine the exact paths that packets should follow usually accomplish this by manually adding static routes to the routing table. Static routes are sometimes undesirable in certain environments. This can only be beneficial on much smaller networks with just a few nodes, i.e. where static routes are predictable and manageable. A main disadvantage is that static routing requires human intervention to execute and/or appropriately respond to changes and updates, e.g. during a maintenance or when there are failures that affect the topology. Static routes (which are often manually managed) are thus impractical for use in larger and more dynamic environments, where their lack of scalability could easily become a major issue of great concern.

On the other hand, dynamic routing uses dynamic processes to learn about and add routes to the routing table. Dynamic routing does not depend on external intervention and is designed to respond automatically (and thus faster), whenever there are topological changes. As a result, dynamic routing protocols are widely used on the public Internet, as well as within private enterprise networks.

Despite the many disadvantages of statically configured routes, they are still in limited use today, e.g. as default routes.

Types of Dynamic Routing Protocols

Classification of dynamic routing protocols can also be based on the kind of algorithms they use to carry out their routing functions. The most common of these include;

- **Distance-Vector routing protocol:** Distance vector routing protocols use two parameters, the distance (e.g. number of hops to reach the destination) and a vector (direction) to determine the route. The number of hops is actually the number of layer 3 devices (routers) that the packet travels through to get to its destination. With this method, the router only knows the distance (or metric) to get to a remote network and the vector (path or interface) to use to get there. Routers using this method do not have an actual map of the network topology. They rely on periodic exchanges with their neighbors to maintain a current topology. Interior Gateway Routing Protocol (IGRP) [191], as well as Routing Information Protocol (RIP) [84], which is one of the earliest routing protocols, are both based on this approach.

- **Advanced Distance-Vector routing protocol:** Advanced distance-vector (or balanced hybrid) is a Distance Vector routing protocol with advanced features that overcome some of the limitations of the original distance-vector protocol. A typical example is the Enhanced Interior Gateway Routing Protocol (EIGRP) [177], which until 2013, was a Cisco proprietary protocol. In 2013 Cisco handed it over to the Internet Engineering Task Force (IETF) for public release as an RFC (RFC 7868) to enable implementation by other vendors as well. A major difference between this protocol and the other DV protocols that use only the number of router hops as metric, is that, it integrates better network features, such as smallest bandwidth on the path and accumulated delay from the packet source to the final destination, in calculating its metrics.

- **Link-state routing protocol:** The link-state routing approach uses the Shortest Path First (SPF) algorithm [62] to compute best paths to all sections of the network. Each router in the topology has a complete map of the whole topology.

  Open Short Path First (OSPF) [145] and Intermediate System to Intermediate System (IS-IS) [156] are two examples of link-state routing protocols. Link-state routers exchange link-state messages with one another to maintain and update their link-state databases (LSDBs). Each router constructs and maintains its own copy of the LSDB, which provides a complete topological view of the whole network, from that router's perspective. The routers use the Shortest Path First (SPF) algorithm to determine the best path to all subnets on the network. The cost associated with the individual links (also known as the link weights) are used for these calculations. The smaller the total cost, the better the path and the more traffic it attracts. Thus, link weights are also used to express preference via specific paths through the network.

- **Path vector routing protocol:** The path vector routing approach exchanges information about the existence of networks and subnets and the path that needs to be taken to reach them. The path information provides the best path to reach the remote network and prevents routing loops. It is, to an extent, similar to the distance vector approach in that it also does not provide a full

topological view from the perspective of the individual routers. However, it also differs from it, by the additional path information that it provides. BGP [169] is a Path Vector routing protocol.

Intradomain versus Interdomain Routing

There are generally two major categories of IP routing protocols; those that operate within an Autonomous System and those that operate between different Autonomous Systems.

Both however share the same fundamental design goals, i.e. learning about routes, choosing the best route if multiple competing routes exist for the same destination network and converging whenever there is a change in the network topology.

- **Intradomain Routing:** Intradomain routing protocols are designed for use within an AS. These protocols offer different features and trade-offs, eventually leading to a diversity of routing protocols. Since routers within an AS are under the full control of their ISP, the ISP can freely select the routing protocol that offers the best trade-offs or which best meets its needs. Trade-offs include, ease of implementation, adaptability, amount of control traffic, convergence speed, etc. OSPF and IS-IS are the two most common intradomain routing protocols used by ISPs. Other intradomain routing protocols include, RIP, IGRP and EIGRP.

- **Interdomain Routing:** Interdomain routing is designed for use between ASes. Routers belonging to different ASes exchange their routing information using interdomain routing protocols. BGP is the only interdomain routing protocol in use today on the Internet.

Although interdomain routing protocols, such as BGP and intradomain routing protocols, such as OSPF and IS-IS, share similar design goals, their main emphases are quite different. BGP's main goals are global reachability and scalability. Its priority is to ensure that all Internet routers learn about all public IP address prefixes that are reachable via the Internet. It thus has to deal with a relatively huge routing table size, compared to the relatively smaller table sizes that intradomain routing protocols have to deal with. The current size of the BGP routing table stands at above 750,000 [96]. This is quite a huge number that warrants scalability, as more and more prefixes are being added, due to continuous growth. At the moment, BGP is the most suitable routing protocol that effectively handles the exchange of such huge numbers of routes between ASes, as is required on the global Internet.

Multipath Routing

Multipath routing is a feature that enables multiple paths in the network to be used when forwarding packets between a given source and a given destination. The distribution of traffic across multiple paths provides better load and resource sharing, which in turn, improve the overall performance of the network. There are two types of multipath routing schemes:

- **Equal-Cost Multipath routing (ECMP)** forwards packets to a single destination over multiple "best paths" of equal metric value (cost). It is a per-hop decision that is limited to a single router. Various routing protocols, including OSPF and IS-IS explicitly allow ECMP routing. A general discussion on equal-cost multipath routing can be found in RFC 2991 [202].
- **Unequal-Cost Multipath routing** occurs when forwarding to a destination is done over multiple paths of different metric values (costs). In this case, a variance value is used to indicate/limit the range of considered metric values. An important condition for this feature is ensuring that the routes via the alternate paths are loop-free. IGRP and EIGRP are two intradomain routing protocols capable of carrying out this check. They thus support unequal-cost multipath routing, in addition to their support for equal-cost multipath routing. The interdomain routing protocol, BGP (or eBGP to be more specific), also supports unequal-cost multipath routing. BGP ensures that its routes are loop-free, by checking all routes received from external ASes, to see if its own AS number is in the AS_PATH attribute and discarding them if there is a match.

Corresponding to the above routing schemes, equal-cost and unequal-cost load-balancing can respectively be achieved when they are used. With equal-cost load-balancing, the load is shared equally between all paths. In the case of unequal-cost load-balancing, the amount of traffic sent across a particular path is inversely proportional to the path's metric value.

2.5.2 Multiprotocol Label Switching (MPLS)

MPLS [175] is an efficient method of forwarding packets between networking nodes, using short, unstructured and fixed-length labels instead of the conventional longest prefix match algorithm. The goal of MPLS is to provide better QoS to connection oriented services, support traffic management that improves network throughput and retain IP-based networking flexibility [185]. MPLS allows Service Provides to connect many different customers on to the same IP network but use label switching to keep their IP traffic separated from each other.

In conventional IP networks, packets that are forwarded across multiple routing nodes undergo a substantial amount of processing delay, as each router first extracts

the layer-3 header to get the destination address, then looks it up in its routing table and finally performs the longest prefix match algorithm to determine the next hop address. With label switching, the extraction is done only once, in the beginning and then mapped onto a value called the label. After a label is assigned, a short label header is added in front of the original layer-3 header and forwarded across the network as part of the packet. Subsequent MPLS nodes no longer need to repeat this analysis. All they have to do, is simply swap the labels accordingly and then forward them based on their *Forwarding Equivalence Class (FEC)*. All headers that map onto the same label use the same next hop.

The MPLS forwarding table lookup process is both less complicated and fast because of the use of unstructured and fixed length labels and the avoidance of the prefix matching overhead.

2.6 Quality of Service (QoS)

The bandwidth aggregate at the edge of the network is often much larger than that of the backbone link that carries the aggregate traffic. When much more traffic arrives at the router's ingress links than could be forwarded out its egress link(s), the router has to make a decision whether to forward them in the order they arrive, prioritize one kind of traffic over another, which packets to discard when its queues are full, etc. Queue scheduling algorithms, such as First In First Out (FIFO), Weighted Fair Queuing (WFQ), Low Latency Queuing (LLQ), etc are used to determine the next packet that should be forwarded out the egress interface.

Quality of Service (QoS) is the tool that networking devices use to differentiate between various classes of traffic in order to prioritize their forwarding as they flow through the device. This priority is usually with regards to bandwidth, delay, jitter and packet loss. For example, some applications, such as non-interactive data backup, require much bandwidth, with delay and jitter remaining less critical. Another application, such as voice, requires less bandwidth, but better (low) delay, no jitter and no packet loss to perform optimally. Yet others, such as video conferencing, require much bandwidth low delays, low jitter and no packet loss to perform optimally. Thus, the goal of QoS is to provide the different types of traffic, different aspects of its QoS feature as is required by each of them to function optimally.

2.7 Network Measurement

Measurement is an essential part of every engineering undertaking. For networks such as the Internet, these measurements are generally driven by three main goals: social, commercial and technical [42]. In order to effectively measure the network and correctly interpret the results, an understanding of its architecture is essential.

Measurements provide objective records and benchmarks about the network's behavior under given conditions. Through measurement of performance metrics, a network's performance improvements or degradation, e.g. resulting from changes (willful or not) can be quantified and analyzed.

2.7.1 Active Measurement

Active measurements involve adding traffic that serves as measurement probes to the network. The added traffic could affect the behavior of the network and thus distort the results of the measurement. For example, to measure the maximum capacity of a link, active probes are continuously sent through it, with increasing packet sizes, until the link becomes saturated, at which point the maximum value is recorded. Since this could be counter-productive, the effect of the active measurement process needs to be kept minimal.

2.7.2 Passive Measurement

Passive measurement on the other hand, is done by observing, capturing and analyzing normal network traffic already generated by other users and applications. The measurement process does not need to generate its own traffic in order to capture the properties it needs to measure. There is one potential problem with this approach though. Since the measurement system depends on traffic that others generate, there might not be enough of the particular type of traffic needed, to accurately capture the intended measurement property.

2.7.3 Hybrid Measurement

Hybrid measurements combine elements of both active and passive measurements in their function. For example, in scenarios that involve actively sending probes through a network and passively monitoring their progress during the measurement session. Such allow the path of the probes to be tracked and entities like intermediate and end-to-end delays to be recorded. This can't be done through active measurement alone. It must however be noted that hybrid measurements often share the same kind of issues that active and passive measurements respectively have.

2.7.4 Common Metrics

Performance metrics are used to predict the performance of a system under certain conditions. The particular system being studied will normally dictate the type of metrics to be selected. To assess the performance of networks, the following metrics are commonly used:

- **Capacity:** Capacity is a measure of the quantity of traffic that a system can handle. It is typically measured in bits per second (bps) or packets per second (pps).
- **Throughput:** Throughput measures the rate at which data is sent through the network, by counting the bytes that are delivered within a specified time window. This time window needs to be selected intelligently in order to capture short-term spikes or drops as well. This might however mean collecting data at much higher rates, thus requiring more system resources and capacity. Depending on the level of sensitivity desired, time windows in 1 to 5-minutes buckets are recommended. Throughput can be measured in bits per second, bytes per second or number of packets per second.
- **Goodput:** Goodput measures the amount of useful application-level data that is transmitted per unit time. It excludes all protocol overhead information, such as packet headers and any other data involved in the transfer process, even retransmitted data.
- **Link Utilization:** The utilization of a link is simply a ratio of the traffic currently being pushed through the link in bps to the link's physical (maximum) capacity in bps, expressed as a percentage.
- **Delay:** Delay is the amount of time taken to transmit a packet from its source to its destination. This is often referred to as end-to-end or one-way delay.
- **Latency:** Latency is an expression of the delay that packets experience while traversing the network. Network latency is measured by means of **Round Trip Time (RTT)**, which is the length of time that surpasses between sending a request from a source system to a destination system until the time the source system receives a corresponding reply from the destination system. In other words, this is the time it takes a packet to go from the source to the destination and back. This encompasses i) the time it takes the packet to travel through the physical links on its path (transport time) ii) the time it takes the packet to go through all intermediate routers on its path (queuing and transmission times) iii) the time it takes the destination to process the packet and send back a reply (destination response time).
- **Packet Loss:** Network packet loss is the fraction of packets that are lost in transit within a specified time interval, expressed as a percentage of the total traffic that was sent within that same time interval. Packet loss gives an indication of the level of congestion in the network or the level of physical impairment in a transmission medium e,g, cable breakage in wired links and magnetic or electrical interference in wireless or mobile connections.
- **Jitter:** Jitter is the measure of the difference in delay that subsequent packets experience while traveling from a common source to a common destination. It is simply the change in latency from packet to packet.

2.7.5 Common Measurement and Monitoring Tools

- **PING:** PING is a common probe tool used for active network measurements [120]. It runs on end-hosts, as well as on intermediate systems such as switches and routers and is often supplied as part of the Operating System (OS) of the device. The ping utility in the source system is used to generate and send Internet Control Message Protocol (ICMP) [163] *echo request* messages to a target destination system. It then starts a timer when it sends off the messages. The target system simply reverses the ICMP headers and sends back the packets to the source system as its corresponding ICMP *echo reply* messages. The source system stops the timer the moment it receives the responses and is now able to deduce the RTT between itself and the target system. A successful receipt of these responses indicates that the target system is connected to and reachable via the network and that it is in a good enough state that permits it to respond. This also indicates a functional network with an active path between both systems.

- **Traceroute:** Traceroute is another ICMP-based tool, which, like *ping* is also commonly used for active network measurements. It uses the ICMP *Time Exceeded* message as the basis for its measurement function. The traceroute tool generates and sends UDP packets to a given target, starting with a minimal Time-To-Live (TTL) value of 1 for the first set of packets and increasing this by 1 for each subsequent set of packets. Intermediate routers that process these packets reduce the TTL by one before passing them on. When they notice that the TTL is zero, they drop the packets, generate ICMP Time Exceeded messages and send them back to the source system with their IP addresses included. The elapsed time between transmission of the UDP packets and reception of the corresponding ICMP Time Exceeded message are recorded.

- **PROBE:** PROBE [20] is a network diagnostic tool, which, like the ping tool, can also be used to query the status of an interface. However, unlike the ping tool, it does not require bidirectional connectivity between the probing and the probed interfaces, but instead needs it between the probing and a proxy interface. The proxy interface can either be on the same node as the probed interface or on a neighboring node, to which the probed interface is directly connected (e.g. via local links in IPv6 networks). The Probe tool uses ICMP Extended Echo functionality (which are disabled by default) to formulate and send its request messages. Thus, for the Probe tool to be used on a device, the ISP (or network operator) first has to enable ICMP Extended Echo functionality on that device and restrict access to it via policies. For security reasons, only configuration options enabled by the ISP, will be accessible to legitimate users.

- **Simple Network Management Protocol (SNMP):** SNMP is an ubiquitous network management tool that provides lots of information about the

operational status of network management elements. It functions via a polling operation, whereby a Network Management Station (NMS) is used to poll and retrieve collected measurement data from different managed elements on the network.

- **Remote Network Monitoring (RMON):** RMON is standardized network monitoring specification that allows various network agents and console systems to exchange network monitoring data. RMON can be set to monitor a specific set of features and poll their data, which are stored in databases known as Management Information Base (MIB). RMON-1 [210] provides link-layer statistics for Ethernet (i.e Ehternet, Fast Ethernet and Gigabit Ethernet) interfaces. It provides the ability to filter and capture packet contents as well as the generation of alerts and alarms when thresholds are exceeded. SNMP can then be used to send such alarms to a central monitoring station. RMON-2 [211] includes MIBs that extend the RMON-1 architecture to include analysis that go way up to the Application layer.

- **Netflow:** Netflow [35] is a flow-based network and traffic monitoring and analysis protocol developed by Cisco Systems. A flow is generally a unidirectional sequence of IP packets that possess the same attributes (i.e. source address, destination address, source port, destination port, layer 3 protocol type, Type of Service (ToS) and switch/router interface). Netflow enables the monitoring, collection and analysis of traffic volumes and flows as they enter or leave interfaces. It can also be used to identify the applications that generate the observed traffic and the proportion of bandwidth that each application is consuming. Netflow has four main components, which enable it to monitor, export, collect and analyze data.

  - **Netflow Monitor** is the component that collects flow information on device interfaces. The monitored data is recorded and stored in cache.

  - **Neflow Exporter** aggregates data into flows that are exported to collectors as flow records.

  - **Netflow Collector** is a central server that collects and stores all flow records sent by the remote exporters in monitored devices. UDP is used for the data transfer.

  - **Netflow Sampler** is used to reduce the load on the device running Netflow. It does so by limiting the number of packets selected for analysis, thereby sacrificing monitoring accuracy for device performance.

  Netflow can also be used for security analysis and accounting/billing. Deeper network insight is thus possible with Netflow than is possible with SNMP. There are similar flow-based technologies developed by other manufacturers, e.g. JFlow [149] from Juniper, Cflow [78] from Lucent and sFlow [179], which

is jointly developed by 3COM/HP, DEll and Netgear. However, none of these is as popular as Netflow.

- **sFlow:** Sampled Flow (sFlow) is a sampling technology for monitoring traffic in data networks containing switches and routers [159]. It provides general purpose sampling at layers 2 through 7. It combines interface counters and flow samples into sFlow datagrams that are sent across the network to an sFlow collector.
- **IPFIX:** The Internet Protocol Flow Information Export (IPFIX) [34] is a network flow standard that was created to develop a common, universal standard of export for flow information from routers, switches, firewalls, and other infrastructure devices. IPFIX defines how flow information should be formatted and transferred from an exporter to a collector.

3

Network Traffic Management

In this chapter, we focus on the traffic management challenges that ISPs face on IP backbone networks. We start with an introduction of some of these challenges and point out how they are impacting performance. We then present some of the common solutions, as well as some of the most impactful methods/techniques/proposals that researchers and operators are using to address and curb these challenges/issues.

Network Traffic Management (NTM) (or Traffic Engineering (TE)) is a collection of techniques that seek to address performance-related evaluation and optimization issues in operational IP networks [14, 38]. It encompasses a design and improvement process, which starts with the evaluation of technological/scientific principles and techniques, with respect to measurement, modeling, characterization and control of traffic and ends with the application/implementation of these principles and techniques on the operational IP network, with the goal of achieving specific performance objectives.

NTM has two major objectives:

i) the timely addressing of traffic-oriented performance requirements.

ii) the enhancement of network resources and network traffic performances, through economical and reliable use of available resources.

Depending on the specific objective and taking the dynamic nature of traffic flows and traffic volumes into consideration, the required timescale to address related issues could range from just a few minutes (e.g. when dealing with spikes and bursts), up to a few years (e.g. when topological adjustments are needed to accommodate growth-forecasts or quell peak demands).

In general, the overall traffic management task includes;

- Capacity planning with the use of traffic load forecast,
- Equipment configuration and management,
- Network usage/load monitoring,

- General traffic engineering and other approaches that help improve network operations and performance [13] [42] [54] [61] [160].

Core networks have different traffic management requirements and challenges than user-facing access networks do. While diverse and easily implementable solutions exist for user access networks, improvements on core and backbone networks are usually more complex and costly.

3.1 Traffic Management Challenges

As has already been mentioned, current traffic growth rates on backbone networks impose mounting challenges on the ISP's ability to control and manage this critical section of its infrastructure. Their business models and the network architecture are both forced to adapt to rising and erratic demands for bandwidth. In addition, the demand for other popular business services (e.g. wholesale and transit, private-lines, mobile backhaul and wavelength switching) continues to grow rapidly as well. These services share the same underlying optical transport infrastructure, upon which the Internet and IP services are built. They also rely on the available backbone capacity to operate efficiently. ISPs continue to add backbone capacity (including enough reserves to absorb unavoidable spikes) as a means to keep up with the high and growing demand.

Augmenting the capacity on backbone networks is a costly undertaking, which unfortunately is also becoming a riskier one [196]. The is because the revenue potential from the carried traffic is increasingly falling short of the associated cost required to augment the capacity. The reasons for this are partly technical, partly architectural and partly organizational. However, they are all rooted in the way backbone architectures have traditionally been designed, built and expanded. Growth and expansion rely on linear scaling methods, such as adding components to increase capacity, although the main issue they want to address is itself often nonlinear in nature. Eventually, a critical point is reached, where, linear scaling only leads to decreasing returns. Better methods for addressing growth on backbone networks are thus needed. Methods that concurrently address multiple dimensions (e.g. the IP and optical control planes), remove architectural boundaries and eliminate the inefficiencies impacting the virtual packet layer, as well as the physical optical layer[1].

3.2 Core Network Architectures

Proposals to address the traffic challenge on backbone networks include changing their architectures, to make them more flexible and less costly. This leads to new

[1] We use the term *physical optical layer* to represent the optical TDM and DWDM layers

architectures that deviate away from the full IP core architecture. Initially, there are predominantly two alternative architectures that result from these proposals; the *hollow* and *lean* core architectures.

- **Hollow Core Architecture:** With Hollow core architectures, expensive core backbone routers are replaced with a transport switching function (an optical transport (OTN) switching layer) that offers much less total cost per bit for a given interface speed.

  The switches create a dense mesh of circuits between each of the edge and peering nodes. However, the management of the optical packet and physical optical layers remain separated from each other. The control-plane integration is also very limited or non-existent. As a result, sharing of topology information between the virtual packet layer and the physical optical layer is not possible. This leaves the routers to know only the routing topology and the optical switches only the optical topology.

- **Lean Core Architecture:** Lean core architectures are adaptations of full IP architectures composed of backbone routers with reduced Network Processing Unit (NPU) functionality or memory. With reduced NPU memory, the routers can only carry out limited routing functionalities, such as learning only internal routes, which forces operators to use less memory-intensive forwarding schemes, such as MPLS instead of IP.

  On the one hand, using routers with limited NPUs sinks the overall cost of deployment. On the other hand, the lean core architecture and its associated Label Switch Routers (LSR) can potentially introduce problems that consume all potential savings. With this type of architecture, all IP services are moved outside of the backbone, to the edge and Provider Edge (PE) routers, eventually converting the backbone to an inner-core. Such changes are usually complex and disruptive to implement, because they often involve re-architecting a network in active operation.

  Just like hollow core architectures, lean core architectures also lack the possibility to integrate the different layers, which continue to be managed separately. Topology information remains isolated within the virtual packet and physical optical levels, therefore limiting the operational efficiency.

Despite the cost-saving advantages offered by hollow and lean core architectures, they still do not address the other challenges associated with isolated packet and optical layers. These include difficulties in monitoring, troubleshooting, provisioning, service velocity, etc. Other architectures are therefore needed, which better integrate these isolated and often independently operated IP and optical network layers (control planes) [45].

3.2.1 Multi-Layer Control Plane

To address the lack of integration between the layers, a consolidated multilayer control plane that works across the packet and optical layers, is created [195].

Generalized MPLS (GMPLS) is a good example of an architecture with a multi-layer control plane. There are two general models of GMPLS operations; the peering model and the overlay model.

- **Peering Model:** In the peering model, a single domain containing both the packet and optical layers is formed. Topology information is shared between Layer 1 and 3 devices. Layer 3 routers thus have visibility into layer 1 transport paths, loads, risk groups, wavelengths, etc. Layer 3 routers are thus able to calculate best paths, request for circuits that meet particular requirements, move and restore failed circuits etc.

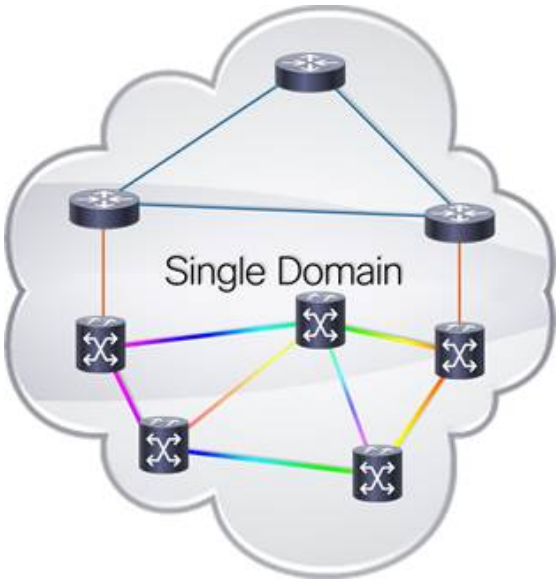


Figure 3.1: GMPLS Peering model (Courtesy of Cisco Systems, Inc)

On the flip side, this model creates 3 distinct issues;

i) The single routing domain it forms, does not respect existing boundaries.

ii) The packet routing and optical switching devices must scale to cope with larger routing domains, adding to memory requirements and computational load across the whole network.

iii) software testing and upgrades must include both the packet and optical layers, which slows down the certification process and complicates deployments.

- **Overlay Model:** In the overlay model, the packet and optical layers remain separated, but use User-Network Interfaces (UNI) to interact with each other. The packet layer acts as a client that requests for information from the optical layer, which acts as a server.

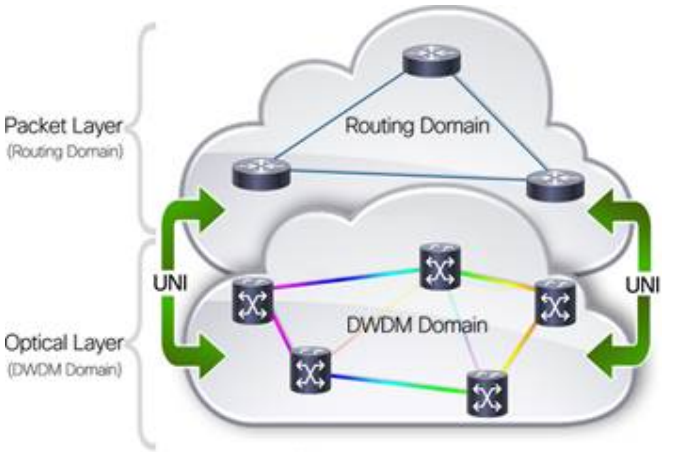


Figure 3.2: GMPLS Overlay model (Courtesy of Cisco Systems, Inc)

Topology information is not shared between the two layers, however, the packet layer can request for circuits to be created and to be removed using the UNI interface. This is the only service that the UNI interface offers.

A main drawback of this model is its lack of information sharing (or only very limited information sharing) between the packet and the optical layers. It provides only a limited improvement in the circuit setup automation. Many of its other functions still need to be done manually, defeating the goal to ease operational complexities and costs.

Neither of the two models presented above provides a satisfactory solution. There is either too much information sharing (with the Peering model) or too little information sharing (with the Overlay model).

3.2.2 Converged Architecture

To simultaneously address current traffic challenges across all layers, without sacrificing one in favor of the other and to improve scalability, increase flexible and reduce cost, a new architecture is proposed. It is known as the converged architecture [45].

In the converged architecture, all components of the backbone network are unified into one architecture, which facilitates effective sharing of information between them.

These components include: the packet layer, the optical layer, the control plane, the management plane, administrative systems, Network Management (and monitoring) Systems (NMSs), as well as Operating Support Systems (OSSs).

Eliminating the boundaries between these layers and components, while enhancing the features and roles of the control plane and NMSs, will aid information distribution and sharing across even organizational boundaries. This creates a more efficient and dynamic backbone network [197].

Although network operation and management is largely economic-driven, with cost, regulations and technology all playing meaningful roles, our focus in this thesis is mainly technological. We however consider the equally important cost and regulatory aspects as given constraints.

3.3 Core Capacity Planning

Planning the capacity of an IP backbone (core) network is an important undertaking that precedes its construction and subsequent upgrades. Various levels of aggregated traffic coming from attached access and peering networks ought to traverse the core without running into any issues. For core networks, *bandwidth* is the most important commodity. The planning process should ensure that enough bandwidth is made available across all sections of the network. All load conditions, including traffic fluctuations, traffic growths and node/link failures, are expected to be satisfied at all times, in order to guarantee committed SLAs.

Effective planning takes four major requirements into consideration:

- The core network topology
- Accurate measurement of current traffic load
- Forecast of future traffic load
- Effective bandwidth provisioning method

Each of these items plays an important role in ensuring that capacity always exceeds the demand and is not (or only minimally) affected by failures. More detailed explanation of these requirements are given in the following sections.

3.3.1 Backbone Network Topology

A common characteristic of backbone topologies are very powerful (core) routers and very fast backbone fiber links. Data exchange between Internet systems often travel across multiple domains/backbones before reaching their final destinations.

Backbone traffic is thus an aggregate of traffic from multiple sources to multiple destinations.

To measure the amount of traffic flowing through the backbone, a corresponding topological representation of the backbone network is needed. The level of details included in the representation (such as AS-level, PoP-level, Network-level or IP-level details) should correspond with the degree of traffic aggregation and the granularity of the intended measurement results.

Peering between providers often occurs in multiple locations, resulting in multiple ingress/egress points, through which traffic could be exchanged. Therefore, it is possible that similar flows through the same backbone network could be using different ingress/egress points. An appropriate topology map is one that contains such necessary details as well.

We presume that each ISP best knows its own topology. Detailed topology information constitutes confidential business secret that can not be freely made available to the public. Although high-level topology information are sometimes made public, they often do not contain the level of details or accuracy corresponding to the real topology. *The Internet Topology Zoo* [124] is an initiative that collects and processes such publicly available topology datasets from ISPs across the world. Internet researchers thus rely on inference as an alternative method to acquire representative topology maps [23, 183, 219]. However, important aspects such as route diversity are often either lacking or incomplete. Improvement is offered by [146], using multiple routers (instead of a single router) in AS topologies, to enhance the accuracy of capturing path diversities from/through the respective ASes.

Factors that affect topological changes and/or traffic flows [167] should also be taken into consideration. For example, until recently, transit backbone networks of Tier1 ISPs carried the bulk of all Internet's inter-domain traffic. However, ever since the insurgence of CDNs, a large portion of this traffic now flows directly between them, hosting/CDN networks and consumer networks [130]

3.3.2 Traffic Demand Measurement

Traffic matrices are essential for most network performance and analysis studies. In section 3.3.1, we explained why representative backbone topology maps are indispensable for accurate traffic measurements. The results of such measurements are stored as elements in a traffic demand matrix. An element denotes the volume of traffic per timescale that was measured between a pair of origin-destination nodes. Nodes in the topology map could represent domains, PoPs, networks or routers.

Measurements over short time intervals, i.e. from a few microseconds to a few minutes, are generally used to analyze performance issues with short timescale properties. Likewise, long timescale measurements, that range from minutes to years are

used to tackle network engineering issues of longer timescales. In practice, different measurement methods are used, as a result of such timescale differences [42, 49, 144, 158, 206].

Accurate measurement of the traffic demand is quite challenging with current methods. Direct measurement is done by gathering and storing important statistics e.g. using SNMP and NetFlow for IP networks or Link State Protocol (LSP) statistics of Label Distribution Protocol (LDP) and Resource Reservation Protocol (RSVP) for MPLS networks. It is not always possible to accurately measure every element of the traffic demand matrix within the same time frame and under the same conditions. Routing policy changes or other changes that affect flows and the paths they use, could introduce inaccuracies or errors that falsify the measurement. Estimates are often used to supplement where measurements are either incomplete, practically not possible or are possible but lead to inaccurate results. A number of these estimation methods are evaluated in [81]. The best estimation approaches capture network and flow dynamics to help eliminate sources of error and improve their levels of accuracy [53, 98, 182].

3.3.3 Traffic Demand Forecast

Forecasting is a key readiness factor when planning to accommodate future growths. Traffic demand forecasts are predictions of future loads in anticipation of such growths. They provide the basis for prior performance evaluation and analysis. Good forecasts depend on techniques that use a combination of historical and current datasets to predict the future loads. However, even the best forecasts are still error-prone by considerable margins.

3.3.4 Bandwidth Provisioning

Provisioning ensures that enough bandwidth is allocated to accommodate demands, while simultaneously guaranteeing that all performance attributes defined in SLAs with customers are maintained. Factors that influence the quantity of the bandwidth to be provisioned, include; expected/unexpected traffic growths, backups in case of failures/maintenances, changing usage patterns and other activities that affect the volume and path of traffic as it flows through the network.

An efficient provisioning approach is essential to avoid wastage and minimize cost. Instead, most ISPs use simple rules of thumb approaches to over-provision backbone links, by as much as 10-folds in extreme cases. Over-provisioning ensures that there is enough capacity in the network to meet demands, especially during peak times and under failure conditions.

The 40% or 50%-rule, requires links to be upgraded when their average utilization exceeds 50% [201]. Another approach recommends normal operation up to an average link utilization of typically 35%, then to upgrade when this increases to averages between 40 and 60% of the link capacity. More recent approaches recommend average utilizations as high as 80 - 90% before upgrading, for traffic made up of a mix of many flows [59].

Several proposals exist that request the change and improvement of the current status quo, including implementing new architectures at the core [44], as presented in Section 3.2.

3.4 Network and Traffic Engineering Approaches

Newer and better approaches are needed to resolve traffic issues on backbone networks. Current approaches focus on solving a single dimension of what is actually a multidimensional problem.

The goal of network engineering is to alter the network to match the traffic flowing through it. On the other hand, the objective of traffic engineering is to alter the traffic flowing through the network, so that it matches the topology. Either approach or a combination of both could be used to obtain an optimized network infrastructure, as will be shown later.

Traffic engineering plays the distinctive role of controlling and optimizing the routing function. The goal is to influence the traffic that flows through the network, so that it is forwarded in suitable ways that satisfy one or more chosen performance objectives. Such objectives include; reduction of average packet delays, avoidance of network congestions, traffic load balancing across multiple paths and traffic rerouting around failed links and devices.

3.4.1 Software and Hardware Upgrades

The Internet is driven by speed; speed of the transport media, speed of the forwarding hardwares (routers and switches), speed of the softwares that drive these hardwares, speed of the applications, speed of the servers/systems that host these applications, etc. Rising speeds to a large extent, together with other important factors, account for the observed good general performance and improved user experience on the Internet.

Network Service Providers are taking advantage of advances in software and hardware technologies to upgrade their infrastructures. They are replacing legacy devices with state-of-the-art devices possessing features that offer superior technical, performance, management and cost benefits.

The manufacturers of these devices work together with researchers to realize and implement such key features, which then get integrated into subsequent software/hardware releases and models. Manufacturers strive to boost performance and confidence in their products by addressing known (software) caveats and releasing frequent bug-fixes and updates that customers are expected to implement as they become available. They work together with their customers to gather feedbacks on such fixes, as well as on new features that the devices offer.

Emerging trends that cause shifts in the way things are done, often also warrant new kinds of services and (sometimes also) special devices to run or support the services. Software-Defined Networking, Virtualization and Cloud-hosted services are trends currently impacting the networking industry in general and ISP's in particular. An appropriate infrastructure needs to be put in place to support the service landscape that the current trend has called into existence. This simply implies hardware and/or software upgrades are needed to get the infrastructure ready. Inspite of all these, the one factor that seems to accompany all major trends on the Internet, is the constantly increasing traffic volumes and flows that ISPs have to deal with.

With regards to networking devices, advancement in chip technology has led to the manufacture of devices that are more powerful and faster in both processing and transmission speeds [148, 194]. Line speeds of up to 100Gbps on a single channel are currently attainable with these new devices [150, 151, 193].

Although the scale of such upgrades offer various technical and business benefits, on the flip side, they sometimes also require high initial investments that in turn, involve longer and tedious justification processes before the budget is granted.

3.4.2 Additional Nodes and Links

The physical topologies of most backbone networks remain unchanged over relatively long periods of time. Addition of new nodes and/or links causes changes to the topology that need careful planning and analysis prior to installation. A more common practice with ISPs is to upgrade existing nodes and links without necessarily changing the topology. Rapid traffic growths have however compelled Providers to upgrade backbone links and/or nodes (routers/switches) at increasingly shorter intervals. Still and all, a point then comes when the topology also needs to grow, especially when the load grows to levels unsustainable using current upgrade means. Adding new nodes/links becomes an inescapable necessity.

Depending on the traffic dynamics, the ISP can decide to only add new links between existing nodes to feed high demands. The new link can therefore be added as:

- a parallel link to existing link(s), thereby boosting the bandwidth between the two nodes, or,

- a completely new link between nodes where none existed before. This adds a new path to the topology, resulting in a new topology.

In both cases, the addition affects IGP routing metrics, which further influence the paths that flows take. Prior evaluation is thus an important and unavoidable requisite.

A lot more needs to be taken in consideration when adding new nodes. Nodes always need to be attached to other nodes. Therefore, new links also need to be planned wherever new nodes are being added to a topology.

Adding new nodes/links is usually a very costly investment for the ISP. On the flip side, falling price per byte turns to reduce (or even negate) the ISP's return on such investment. ISPs therefore follow the simple and less expensive approach of preferring the shortest possible physical distances when considering where to add/move links/nodes in the topology.

3.4.3 Change of Traffic Matrix

Traffic matrices provide clues on why the traffic distribution in a network is the way it is and the effects that changes would have. TMs are subject to change with changes in usage patterns and trends. This means factors affecting today's TM might not necessarily be the same factors affecting tomorrow's. To achieve specific goals, the TM could be influenced or changed by changing the routing policies that affect them.

A typical scenario involves traffic demands with flows between domains, i.e. flows that transit via dedicated ingress links through to a set of egress links on an ISP's backbone. The demand model allows prediction of how changing the internal routing impacts the distribution of load on the ISP's backbone links [53].

Another approach uses Valiant Load Balancing (VLB) to support all possible traffic matrices [220]. VLB is based on a fullmesh topology supporting equal load balancing across 2-hop paths between any ingress and egress node of the backbone network. In the first stage, traffic arriving at each node is divided into equal parts and sent to each of the backbone nodes, irrespective of the final destination. In the second part, each intermediate node sends its part of the traffic to the ultimate destination.

3.4.4 Flow Rerouting

A further way to influence the traffic that flows across backbone networks is to alter their path, by means of flow rerouting. Flow rerouting occurs either as a response to changes in network conditions, changes in security and routing policies, preparation for network maintenance or as a means of achieving a desired Quality of Service (QoS) goal.

A big advantage of flow rerouting is its applicability/suitability in resolving issues of smaller timescales. For example, it can be used to avoid bottlenecks, reduce congestions, balance traffic loads and bypass failures in a matter of minutes.

3.5 Network Optimization

A bulk of the problems faced by ISPs could be seen as *optimization problems* involving decision making, for example, on how and when to upgrade their network to increase its capacity [83].

Generally, optimization is either minimizing or maximizing a given function relative to some set of available choices in a given situation [173]. Researchers and network operators use optimization theory to study network behaviors, analyze their effects and optimize the use of available resources. The optimization problem is defined as a computational problem in which the objective is to find the best of all possible solutions [59].

The parameters to decide upon are called *decision variables*. Only in rare cases are these variables permitted to take on any value from $-\infty$ to $+\infty$. Their values are often instead limited by *variable bounds*. The decision to be made usually depends on multiple *input parameters* that are either given or first need to be determined, before the decision is made. The *objective function* is a function of the parameters and variables. It has to be minimized or maximized via the optimization process. In doing this, certain restrictions have to be defined and maintained. These restrictions are called *constraints*.

With regards to traffic engineering in a network, optimization seeks the best way to route traffic through the network in order to attain stated objectives, while honoring defined constraints.

3.5.1 Routing Limitations

IP routing is purely destination-based forwarding, with the chosen path being the one that offers the best (smallest) total metric (cost). However, the criteria used to determine the metric is lacking in some practical aspects. Taking the link-state intra-domain routing protocol, OSPF, as an example, although calculation of its metric employs link bandwidth, it completely ignores the link load, which is of a better significance. On the other hand, link load is a very dynamic property. Including it in the metric's calculation would introduce a volatile property that could affect network stability and performance. This is because any change in the load would trigger routers to recalculate new metrics, update their neighbors and then wait for convergence. This could eventually result in the selection of different paths for the

same subnet, as well as cause route flapping and instability, which are detrimental to the network's general performance.

Eventually, since only the best paths are chosen, this could lead to over-utilization (congestion) on some links while others remain minimally used or are not used at all. Optimization (as we shall show in Chapter 6) therefore presents a better alternative that could be used to redistribute traffic flows within a network and minimize the maximum load on individual links.

3.5.2 Network Graphs

Graphs are mathematical structures that represent pairwise relationships between objects. Representing a problem as a graph offers a different point of view on the problem that can make it much easier to solve. They find application in many business and engineering domains, including network optimization. Graphs are made up of vertices (nodes) and edges (links) that connect the vertices.

Formal Definitions

Formally, a graph G is defined as G = (V, E), where V is the set of all vertices and E is the set of all edges in the graph. Other attributes of graphs are defined as follows:

Root vertex/node: The root vertex is the ancestor of all other vertices in a graph. It therefore has no parent of its own and is usually the access point into a graph.

Leaf vertex/node: A leaf vertex is one without any successor. It can have many incoming edges, but no outgoing one.

Simple graph: A simple graph has no self-loops and has only a single edge between any two vertices.

Undirected graph: An undirected graph is one in which all the edges are bi-directional.

Directed graph: A directed graph is one in which all the edges point in a single direction only (uni-directional).

Weighted graph: A weighted graph has each of its edges assigned an associated weight or cost. This weight is given by the function $w : E \to \mathbb{R}$.

Structural Properties

Some fundamental structural properties of the graph G=(V, E) are described as follows:

Connectivity: A graph is *connected* when it is possible to reach a vertex from any other vertex in the graph. It is *strongly connected* when there is a direct connection between any two vertices in the graph (fullmesh topology). A graph that is *disconnected* can be split up into a number of connected *components*.

Adjacency: A finite graph can be represented using an *adjacency matrix* or an *adjacency list*. An adjacency matrix A is a 2-dimentional $|V|$x$|V|$ binary matrix, where $|V|$ is the number of vertices in the graph. An element A_{ij} has the value 1 if there is an edge from vertex i to vertex j, else $A_{ij} = 0$. For a weighted graph, the value of A_{ij} is that of the corresponding weight or cost.

An adjacency list is a more vertex-centric way of representing the same information about the graph.

Degree: In an undirected graph, the *degree* of a vertex $deg(v)$ is the number of edges incident on the vertex. In a graph with n vertices, $deg(v) \leq n - 1 \quad \forall v \in V$. In a directed graph, the *indegree* $deg^-(v)$ of a vertex is the number of edges coming into the vertex and the *outdegree* $deg^+(v)$ is the number of edges leaving the vertex.

Walks: A walk of *length* k is a sequence of alternating vertices and edges, such as $v_0, e_1, v_2, e_2, ..., e_k, v_k$. Each edge e_i is given by $e_i = \{v_{i-1}, v_i\}$. Walks can have repeated edges. A walk is *closed* if its starting and ending vertex are the same, i.e. if $v_0 = v_k$. Else, it is considered *open*. A **trail** is a walk with no repeated edges.

Paths: A path is an open trail with no repeated vertices. A shortest path is the minimum path connecting any two vertices.

Topological distance: The topological distance d_{ij} between vertex i and vertex j is the number of edges in the shortest path connecting both of them.

A *distance matrix* D is a $|V|$x$|V|$ matrix with D = (d_{ij}), where d_{ij} is the topological distance between vertex i and vertex j.

Cycles: A cycle is a closed trail, where no other vertices are repeated apart from the one where it starts/ends.

Trees: A tree is an undirected graph in which any two vertices are connected by one and only one path. In a graph with n vertices, a tree is an acyclic graph with n-1 edges. In a graph, each vertex may have one or more parent. In a tree, each vertex has only one parent, except for the root vertex that has no parent.

Some of the above defined properties will be applied in our study and analysis of the P2P overlay in Chapter 4, as well as in formulating and solving the optimization problem in Chapter 6.

3.5.3 Modeling and Solving Optimization Problems

To solve optimization problems, one starts with first identifying the exact problem, stating the variables, the constraints, the objective function and the parameters. Next, the objective function and constraints are formulated as mathematical models. Thereafter, they are solved using standard approaches, such as those mentioned below. To detect if there are potential issues with the model, small problems are often first solved and their solutions carefully analyzed. Only after the correctness of the solution has been confirmed and the confidence enhanced, should more complex problems be attempted. Since input parameters can sometimes be uncertain or incorrect, sensitivity analysis is usually done to find out how sensitive the solution is to changes of input parameters.

The two most popular mathematical programming approaches used to model and solve optimization problems are *Linear Programming (LP)* [147] and *Integer Programming (IP)* [26], respectively [6]. While decision variables in LP problems are allowed to be continuous (or fractional) in value, in IP they take on discrete integer values. Further variants of IP are provided below.

In linear programming, the mathematical expressions for the objective function and the constraints are all linear. Linear programming is the most widely used method of constrained optimization. One seeks to find a set of values for the continuous variables $(x_1, x_2, ..., x_n)$ that minimizes or maximizes a linear objective function z, while satisfying a set of linear constraints (in the form of simultaneous linear equations and/or inequalities). LP problem definition and modeling can involve plenty of variables and equations, as much as millions of variables and hundreds of thousands of constraints [28]. In general, an LP problem can be mathematically expressed as follows [26]:

$$\text{Maximize } z = \sum_j c_j x_j \tag{3.1}$$

$$\text{subject to } \sum_j a_{ij} x_j \leq b_i \qquad (i = 1, 2, ..., m) \tag{3.2}$$

$$x_j \geq 0 \qquad (j = 1, 2, ..., n) \tag{3.3}$$

An integer programming problem is an LP problem in which at least one of the variables is limited to integer values only. In a *pure* IP problem, *all* the decision variables *must be* integers. When the variables are limited to values of either 0 or 1, the type of IP involved is called a Binary Integer Programming (BIP).

A *Mixed-Integer Programming (MIP)* optimization problem is one in which some of the decision variables are real-valued (i.e. can be fractional) and others are integer-valued (i.e. can take on only integer values) [26, 213]. The model is therefore

referred to as "mixed". When the objective function and constraints are all linear, the model is called a Mixed Integer Linear Programming (MILP) model. When nonlinear variables are involved, the model is referred to as Mixed-Integer Nonlinear Programming (MINLP) and is usually more harder to solve. In most cases, MIP is commonly used to mean MILP. In general, the mathematical formulation of an MIP optimization problem is of the form:

$$\text{Maximize } z = \sum_j c_j x_j + \sum_k d_k y_k \tag{3.4}$$

$$\text{subject to} \sum_j a_{ij} x_j + \sum_k g_{ik} y_k \leq b_i \qquad (i = 1, 2, ..., m) \tag{3.5}$$

$$x_j \geq 0 \qquad (j = 1, 2, ..., n) \tag{3.6}$$

$$y_k = 0, 1, 2, ... \qquad (k = 1, 2, ..., p) \tag{3.7}$$

It should be noted at this point that the input parameters $(c_j, d_k, a_{ij}, g_{ik}, b_i)$ may be positive, negative or zero. The above set of expressions can also be stated in matrix notation as follows:

$$\text{Maximize } z = \mathbf{c}^T\mathbf{x} + \mathbf{d}^T\mathbf{y} \tag{3.8}$$

$$\text{subject to } \mathbf{Ax} + \mathbf{Gy} \leq \mathbf{b} \tag{3.9}$$

$$\mathbf{x} \geq \mathbf{0} \tag{3.10}$$

$$\mathbf{y} \geq \mathbf{0} \text{ integer} \tag{3.11}$$

where m = number of constraints
n = number of continuous variables
p = number of integer variables
$\mathbf{c}^T = (c_j)$ is a row vector of n elements
$\mathbf{d}^T = (d_k)$ is a row vector of p elements
$\mathbf{A} = (a_{ij})$ is an m x n matrix
$\mathbf{G} = (g_{ik})$ is an m x p matrix
$\mathbf{b} = (b_j)$ is a column vector of m constants (or right-hand-side column)
$\mathbf{x} = (x_j)$ is a column vector of n continuous variables
$\mathbf{y} = (y_k)$ is a column vector of p integer variables

When $n = 0$, there will no longer be any continuous variables $\mathbf{x}$. The MIP therefore becomes (reduces to) a pure integer program. Also, when $p = 0$, no integer-restricted variables $\mathbf{y}$ will exist and the MIP reduces to a linear program. An LP can also be achieved when the integer requirements in a given MIP are relaxed (or ignored). The LP that stems from this is called the *LP relaxation* of the given IP. Unlike LPs

containing only $\mathbf{x}$ variables, this LP relaxation contains both $\mathbf{x}$ and $\mathbf{y}$ variables and treats $\mathbf{y}$ as a vector of continuous variables.

4

Managing P2P Traffic via Collaboration

This chapter specifically deals with P2P systems and the traffic management challenges attributed to their disruptive nature and high bandwidth consumption affinity. We start with an introduction of P2P systems, then elaborate on their kinds and uses. We then point out some of their strengths and weaknesses, including proposals and contributions from the research community on how to deal with some of these weaknesses and issues. Lastly, we present our proposed solution, the Oracle service, showing how it effectively functions as an enabler for ISP and P2P collaboration. We present multiple simulation studies and analyses, which show the numerous benefits of the service to both the ISP and the P2P user.

In Chapter 1, we saw that Internet backbone traffic has grown to gigantic proportions over the last decade [108]. We also saw that its steady increase poses persistent management challenges to ISPs.

Since network operators always have to plan appropriately well ahead of time, resources for growths are usually made available via short, mid or longterm planning. However, events and trends that suddenly cause significant spikes and increases in traffic volumes and flows, also call for quicker shorter-term management responses. Often, such changes cannot be ignored or postponed, especially when they directly or indirectly impact other services. Peer-to-Peer file sharing is an example of such a phenomenon that warranted both short and long term responses. It accounted for more than 50% of the Internet's backbone traffic [114] [136] at the peak of its popularity. Although this is no longer the case, P2P traffic is still responsible for a significant fraction of Internet traffic in some regions. For example, as recently as 2016, BitTorrent[1] [37] accounted for the highest fraction of uplink traffic for fixed access connections in North America. Its share stood at 18.37%, compared to only 13.13% and 10.33% respectively for Youtube and Netflix, in the same category [207].

[1]BitTorent is currently the most popular P2P application protocol used for file-sharing.

4.1 Peer-to-Peer (P2P) Systems

Peer-to-peer, as it is known today, basically denotes a type of distributed computing system, in which members (peers) come together to share such resources as, content, storage or CPU processing power. In a P2P system, each *peer* represents a participating end-host that functions both as a client and as a server to other peers. Peers connect with other peers to form neighbor-relationships and establish a logical overlay network at the application layer. Although communication between neighboring peers appears to occur directly at the overlay layer, in reality, their corresponding underlay nodes might be many hops apart. Each peer in the overlay network represents a forwarding node, similar to routers in the physical underlay network. However, the overlay is used mainly for peer discovery and for indexing, while all data exchanges between the peers occur via TCP using the physical underlay network.

4.1.1 Unstructured P2P Systems

Unstructured P2P overlays are those that are established arbitrarily, i.e. without any defined global structure. Peers rely on their adjacent neighbors for packet delivery to other peers. Message propagation occurs via flooding and random walks [21]. Although the lack of a mandatory global structure eases the establishment of the unstructured overlay network, the same can also lead to a suboptimal overlay topology that is still robust under high churn. There is thus room for optimization, e.g. via localization, in various segments of the overlay network. The most prominent example of an unstructured P2P system is Gnutella [123]. Our P2P studies in this thesis are based on *Gnutella version 0.6* implementation.

4.1.2 Structured P2P Systems

Overlays in structured P2P systems are established according to a predefined criteria. The most common type uses a *Distributed Hash Table (DHT)* to assign content ownership to particular peers. In this case, hash functions are used to map peers and to reference shared content, onto a common identity space [119]. This identity space consists of (key, value) pairs that are stored in a database, allowing participating peers to retrieve any value by reference of its associated key.

4.1.3 Performance Challenges

The very nature of P2P systems that brings advantages such as robustness, scalability and high content availability, also accounts for some of its major weaknesses, such

as high churn, high signaling traffic, free-riding, etc. A number of these challenges are further elaborated below.

- **Bootstrapping:** New peers that need to join a P2P overlay first need a list of already existing and preferably online peers to select the ones to establish connections with. A base list can be supplied as part of the client application, which is updated (via download) during first-time use or a preferred list is gotten from a third-party and then manually added to the configuration file. Irrespective of how the list of peers is obtained, potential neighbors are selected from it, either at random or based on some preferred criteria, such as proximity [22], role, e.g. as *Supernode* [135] or influence, as in the *Swarm Intelligence Approach* [87]. The number of neighbor-relationships that a new peer can establish is finite. It is dependent on many factors, such as the size of the list, how current it is and the number of its peers that are also online at the time of connection.

- **Unpredictability:** The performance of the P2P network is largely influenced by the nature of its overlay architecture, i.e the number of participating peers, their contributed resources and how they interconnect with each other. A common issue with P2P networks is the dynamic nature of the peer membership and neighbor relationships. The system is based on voluntary participation, meaning, peers join and leave the overlay network whenever they want. However, this variability also introduces unpredictability in the system, causing its reliability, scalability and even performance to be impacted, when large numbers of peers are simultaneously affected.

- **Efficiency:** Earlier unstructured P2P systems, such Gnutella, use flooding to forward queries from a requesting peer to other peers in the network. A peer formulates a search query and sends it to all its directly attached neighbors. These neighbors check in their databases and respond accordingly, in case there are hits. They then forward the same query string to all of their own neighbors. These next level of neighbors also search locally and respond accordingly before forwarding the search string yet again to their own set of neighbors, and so on. Such flooding feeds on available bandwidth. As a result, a single search can produce a multi-fold increase in overhead traffic.

  The way the overlay topology of a P2P network is established also affects the cost and efficiency of communication in the P2P system. Overlays constructed via random connections between the peers often possess little or no correlation with the physical routing underlay [5, 91, 133, 134].

- **Free-riding:** The main reason why P2P systems were created (and became so popular), was to enable participating users to freely share their resources. However, it was quickly discovered that most participating peers are selfish [180]. They consume the system's shared resources but contribute little or nothing to it [80]. Such peers are called *free-riders* [1]. Free-riding is a common

phenomenon in file-sharing P2P systems, where studies show that only a small percentage of the peers contribute a large percentage of the shared files [94, 176].

4.1.4 Improvements

There have been several improvement proposals/implementations to the original P2P applications and systems. Starting with bootstrapping, [125] analyzes existing methods used for bootstrapping, then makes a number of proposals for their improvement. For unstructured P2P systems, [73] presents an approach, which does not rely on host or web caches, but on DNS-based profiling.

The structure of the Gnutella overlay was changed from a single flat layer to one with two hierarchical layers. Two categories of peers were introduced, *ultrapeers* and *leaf-nodes*. Ultrapeers belong to the top hierarchical level. They are characterized by long session lengths and high content availability. They are also often highly connected with many other peers. Leaf-nodes belong to the lower level. They join the overlay by connecting to ultrapeers and often have much shorter session lengths.

The method that is used to propagate queries across the Gnutella overlay has also been improved. It now uses targeted flooding instead of controlled flooding [189]. In addition, information from connected neighbors and pong (response) messages are cached and subsequently used to respond faster to similar queries without needing to flood the network again.

Many proposals for change/improvement, including incentives affecting user behavior have been made [52, 139, 218]. Some of these have already been partly integrated into P2P applications and systems to discourage free-riding and improve fairness and content availability. Despite these initiatives, free-riding and content availability still remain major issues in P2P systems until date.

4.2 The Oracle Service

No one knows a network like the ISP that owns and operates it. Some applications, such as P2P, require this knowledge in order to operate more efficiently. For example, knowledge of the network could help peers built more efficient overlays that align properly (better) with the ISP routing underlay. This will further improve overlay routing, as well as search and download performances. However, no ISP readily hands out information about its network to third parties, because of business and security reasons. Applications therefore attempt to infer network conditions themselves, which often produces less than accurate results and much traffic overhead.

Notwithstanding, the ISP's interest in maintaining an efficient and high-performing network, calls for its involvement in resolving such issues. After all, improved network efficiency and performance benefit both the P2P systems and the ISP. So, can P2P systems and ISPs cooperate to achieve a win-win solution for both parties? We say "Yes!".

Taking all of the above into consideration, we thus propose a solution that:

- encourages and enhances P2P-ISP collaboration
- resolves/minimizes the mismatch issue
- boost P2P and network performance
- is simple and cost-effective to implement/operate
- reduces inter-AS traffic associated with P2P to manageable levels

We propose an ISP-offered free service, which we call the *Oracle*. It enables peers to make informed and better choices about potential neighbors to connect with when bootstrapping and potential sources to download content from after a search resulting in multiple hits.

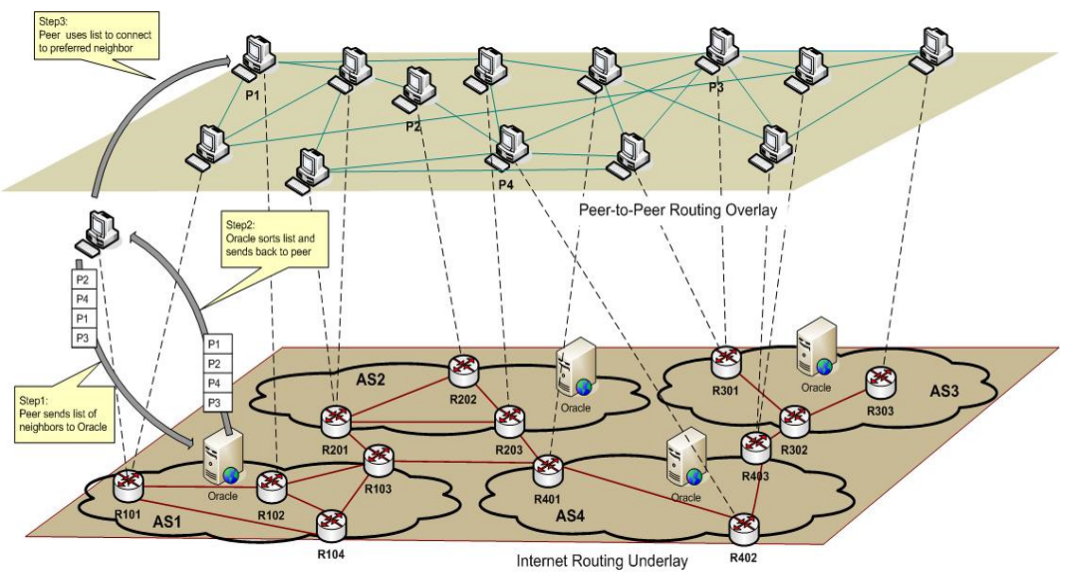


Figure 4.1: Collaboration using the Oracle service

The principle behind the Oracle service is quite simple. As an ISP-hosted service, it has access to detailed information about their network and the connection details of their end-users, such as, each user's respective location, bandwidth and link delays. With such accurate knowledge of the network and its dynamic conditions, the

Oracle service is best positioned to assess potential peers and rank them according to different preferences or criteria, such as,

i) member of local or remote AS

ii) distance to edge of the AS

iii) AS hop count based on BGP metric

For peers belonging to the same AS as the Oracle, it can further rank them according to:

i) bandwidth (connection speed)

ii) link delay

iii) proximity

iv) current (operational) status, e.g. available bandwidth and delay

When peers use the Oracle service, e.g. when bootstrapping to join the overlay or to download from potential peers after a search query, they follow the 3-step approach shown in Figure 4.1, as follows:

- **Step 1**: Send the unsorted list of potential peers to the Oracle server, optionally indicating the desired ranking criteria.
- **Step 2**: The Oracle server sorts the list accordingly and sends back to the peer.
- **Step 3**: The peer uses the ranked list to connect to or download from neighbors that are preferably ranked at the top of its sorted list.

Since peers are not compelled to use this free service, the system is designed in such a way that, it offers great and provable incentives that attracts even the most skeptical users to at least test it. Peers that use the Oracle service benefit in the following ways:

- Only a little change is needed by peers to use the service
- Peers often need to make informed decisions that require knowledge of the underlay network. They no longer need to measure or infer these themselves, but can simply rely on the Oracle service for this
- Improved user experience. A correlated Overlay-Underlay and proximity between neighboring peers, help avoid congestions at inter-AS exchange points, thereby boosting throughput and reducing latency for the peers.

By sending an unsorted list to the ISP service and getting back a sorted one, no further information is released about the peers, apart from that which the ISP is already in possession of. This, in effect, handles any privacy concerns the users might have.

On the other hand, by offering the free Oracle service as an incentive to peers, the ISP benefits in the following ways:

- Without the service, the disruptive nature of P2P traffic will continue to be a bane. The Oracle service effects a more correlated overlay-underlay by influencing how peers connect with each other, which in turn influences neighbor relationships and how traffic flows between them.
- Analysis of P2P traffic flows show that they unnecessarily cross AS boundaries many times over, despite the presence of the same content within their same AS. ISPs prefer to minimize cross-boundary traffic, in order to prevent increased transit charges. By offering the Oracle service which addresses their cause, the ISP is able to regain control of a substantial fraction of the cross-boundary traffic.
- Being able to manage a large fraction of disruptive traffic, recreates room for fair usage alongside other applications.

4.3 The SSFNet Simulator

The name **SSFNet** was formed by combining **SSF** and **Net**, each of which represents a major component of the SSFNet modeling and simulation software. SSFNet is used to model and simulate complex large-scale IP networks and offers packet-level granularity.

4.3.1 Scalable Software Framework (SSF)

Scalable Software Framework (SSF) is a standard-based modeling language. It is used to create object-oriented models of various elements used in a simulation. The SSF environment has 5 fundamental classes; Entities, Processes, Events and In-Channels and Out-Channels [18].

- **Entities** are objects with the ability to possess processes and channels, which enable them connect with each other. An entity can send and receive data within the simulation environment and can be monitored to take account of its processes and data transactions.

- **Processes** control the request and generation of information by entities. Processes that belong to different entities can run simultaneously, as a result of an implemented fairness policy that prevents a single process from running more than once during the same simulation timeframe. A process can be in one of the following states; ready to run, running, suspended or waiting for a resource. There is a scheduling procedure within SSF that schedules processes that are ready to run. Suspended processes that are waiting for a specific simulation timeframe have priority over those waiting for resources.
- **Events** control the simulation run. They simulate data traffic and control how entities handle the same. Events can be saved and be released during processing, making monitoring possible. They can also use aliases to create pointers to other events.
- **inChannels** are like interfaces of an entity, through which data is received. The In-channel of one entity connects to the Out-channel of another entity, from which it receives the events that it needs to process.
- **outChannels** are interfaces of an entity through which data is sent. Data that is produced by processed events and need to be sent to other entities, get sent via the Out-channel.

4.3.2 SSFNet Overview

SSFNet is a collection of Java SSF-based components used to model and simulate Internet protocols and networks at and above the IP packet level of detail [40]. Its main classes, with which basically all Internet models can be created, are organized under two major packages; the **SSF.OS** package and the **SSF.Net** package.

The SSF.OS package is used to model hosts and operation system components, such as protocols, while the SSF.Net package is used to model network connectivity and to create node and link configurations. Both packages help hide the details of the discrete event simulator, causing it to implement the protocols just like it is done in real OSes.

The SSF.OS Package

The main classes in the SSF.OS package are:

- **ProtocolGraph:** defines the protocol used in a host
- **ProtocolSession:** defines the methods of communication that protocols use
- **ProtocolMessage:** defines the packet used in the ProtocolSession to carry simulated data

This package also contains Internet protocol models built on top of the base SSF.OS model, e.g. SSF.OS.IP, SSF.OS.TCP, SSF.OS.UDP, SSF.OS.OSPF, etc, that are used to model IP, TCP, UDP and OSPF protocols respectively.

The SSF.Net Package

The main classes in the SSF.Net package include:

- **Net:** models a network. It loads the model from a DML file and controls all instances of the model.
- **Host and Router:** models a network host as a derivative of SSF.OS.ProtocolGraph, with added networking attributes. Models a router as a special host with multiple NICs.
- **NIC:** models network interfaces for hosts and routers.
- **Link:** models link-layer connectivity between attached host and/or router interfaces.

Domain Modeling Language (DML)

Domain Modeling Language (DML) is a high-level model description language that uses standardized syntax. The syntax specifies a list of attributes (key-value pairs) that can be stored in ASCII readable/writable files. The DML package included in SSFNet aids it in describing and configuring models. All derivative frameworks created on top of the SSF API are able to use the DML package to configure models. The format used for configuration is; key followed by the value in brackets, which indicate the start and end of the value, i.e.

```
key [value]
```

In case of multiple key-value pairs, spaces or carriage returns could be used to separate them, as shown below.

```
key1 [value1] key2 [value2]
key3 [value3]
```

SSFNet models can individually configure and instantiate themselves by querying DML-formated files from the network.

4.4 Collaboration within a P2P Simulation Environment

In this section, we look into some fundamental aspects of the ISP/P2P collaboration offered by the Oracle service. We start by implementing the Gnutella P2P protocol in the SSFNet simulation environment, then design and model a representative Internetwork consisting of multiple domains and mix of Tier1, Tier2 and Tier3 ISPs. We distribute end-hosts within each AS according to their category (see Table 4.1). Each host uses the Gnutella P2P protocol to join the overlay and become an active peer.

We use this setup to assess the Oracle's ability to function as an enabler of ISP/P2P collaboration, rectifier of the P2P overlay/ISP underlay mismatch, enhancer of ISP performance and improver of the general end-user experience.

4.4.1 System Design

We use the SSFNet software to model and simulate the collaboration environment containing ISP underlay and P2P overlay infrastructures.

Property	Tier-1	Tier-2	Tier-3
# AS	1	8	16
# Routers per AS	2	2	2
# hosts per AS	360	40	20
Host interface speed (Mbps)	1000	100	10

Table 4.1: Network Properties

In total, there are 25 ASes, 50 routers and 1000 P2P hosts in the network, distributed as shown in Table 4.1. Taking memory constraints into consideration, we limit the size of the network by using only 2 routers per AS. One router is dedicated to inter-AS connections, while the other serves as the intra-AS router, used to attach end-hosts to the network. Link delays between Tier-1 and Tier-2 ASes and between Tier-2 and Tier-3 ASes are set at 2 msecs and 10 msecs respectively.

Peers that join the overlay topology, take on one of two possible roles, *leaf-node* or *ultrapeer*. Leaf-nodes establish connections to a minimum of 2 and a maximum of 4 ultrapeers. Ultrapeers have at least 10 connections to other peers. They stop accepting connection requests when the count reaches 45. The number of files a peer can share is uniformly distributed between 0 and 100. Peers stay online for at least one second and at most 1500 seconds. Once they go offline, they only rejoin after a period between 1 - 300 seconds. This adds the effect of churn to our experiment. Peers functioning as leaf-nodes can transcend to ultrapeers only after haven been online for at lest 600 seconds.

A peer uses its locally saved hostcache to establish connections with other peers (potential neighbors). The hostcache is simply a list of peers that might have been seen on the network in the past, but with no guarantee that they are online at the time of connection establishment [82]. The Oracle service can help a peer sort its hostcache, according to proximity preference, which biases the establishment of neighbor relationships to favor peers in close proximity to each other. The Oracle uses the following algorithm to sort the list it receives from a peer:

i) First, identify peers in the same AS as the requesting peer and place them at the top of the sorted list

ii) Then, use AS-distance to sort the rest of the peers not in the same AS as the requesting peer

We use the above setup to run three separate experiments based on the following cached file-sizes:

- 1000 and not using the Oracle service
- 100 and uisng the Oracle service
- 1000 and using the Oracle service

All three experiments have the same number of queries and similar response success-rates.

We run multiple test simulations, for variable lengths of time and notice that very little changes occur beyond 5000 seconds. So, we settle with 5000 seconds for each simulation run.

The results and analyses of the simulations are presented in the following sub-sections.

4.4.2 Graph Structural Properties of the Overlay

Topology Visualization

We investigate the topological impact of using the Oracle service and use visualization to compare the P2P overlay structure when no Oracle service is used with the case when the Oracle service is used.

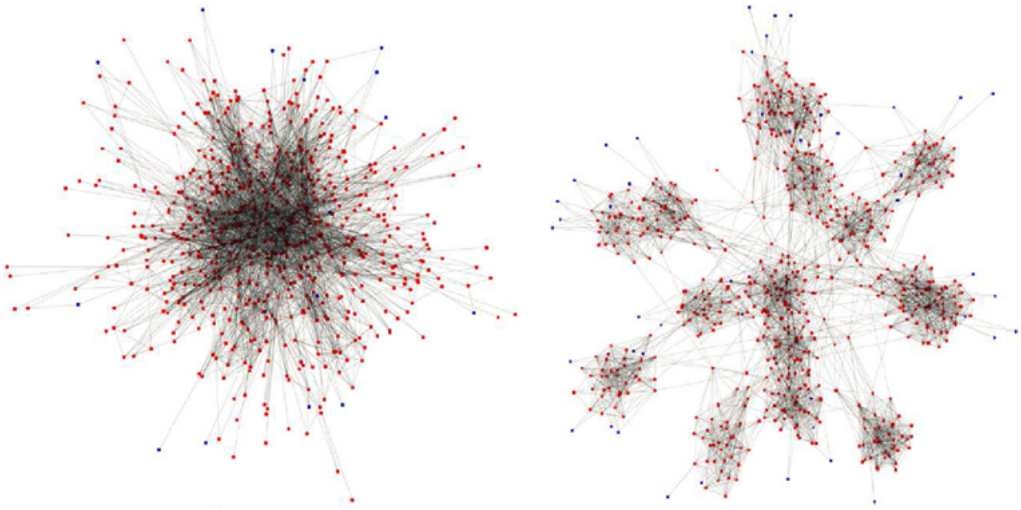


Figure 4.2: Topological Visualization of the P2P Overlay

After the initialization phase (which is about 500 seconds), we let the simulation run again for a reasonable time and then start sampling the overlay topology to capture all peers that are online at the time of sampling. Links are drawn between peers that share neighboring relationships. We then use the visualization library from yWorks [77] to convert these relationships into the structural hierarchical format shown in Figure 4.2. The overlay using the Oracle service, Figure 4.2b portrays a structural resemblance to its underlay. Most links are formed between peers in the same AS, visualized as areas of thickly populated dots and lines. Only a relatively small number of links connect to peers in external ASes. Such a correlated structure is completely absent in the overlay not using the Oracle service, as shown in Figure 4.2a.

Graph Diameter

The diameter of a graph (or network) is the greatest distance (counted in hops) between any two nodes of the graph (or network). Since we use the same underlay topology to test the 3 cases, the AS diameter remains the same (4 hops) in all 3 cases. We therefore compare the diameters of the more dynamic overlay instead. We observe that when the Oracle uses a cache size of 100, its diameter ranges between 6 and 8 hops, compared to only 5 to 7 hops, when not using the Oracle. The range increases to 7 and 12 hops, with an an average of 9.2 hops, when the Oracle uses a list-size of 1000.

Graph Connectivity

Without a central management system, it is possible for the Gnutella overlay to exist as several disjoint overlays [176]. We therefore check if using the Oracle service to bias neighbor selection could provoke this effect. We sample the network after every 500 seconds and check if any of the samples contain split components. None of the samples in all 3 cases contains overlay disjoints or split components, leading us to conclude that using the Oracle service does not negatively impact overlay connectivity (or cause disjoints).

Node Degree

The node degree of a peer is simply the number of links it has to adjacently connected peers. Although node degree is an important topological property of any network, in P2P systems it has a unique significance because of a node's triple role, as client, server and router respectively. The higher the node degree, the better connected the network is. We thus use it as a metric to investigate the Oracle's impacts on the P2P Overlay's structural/topological properties.

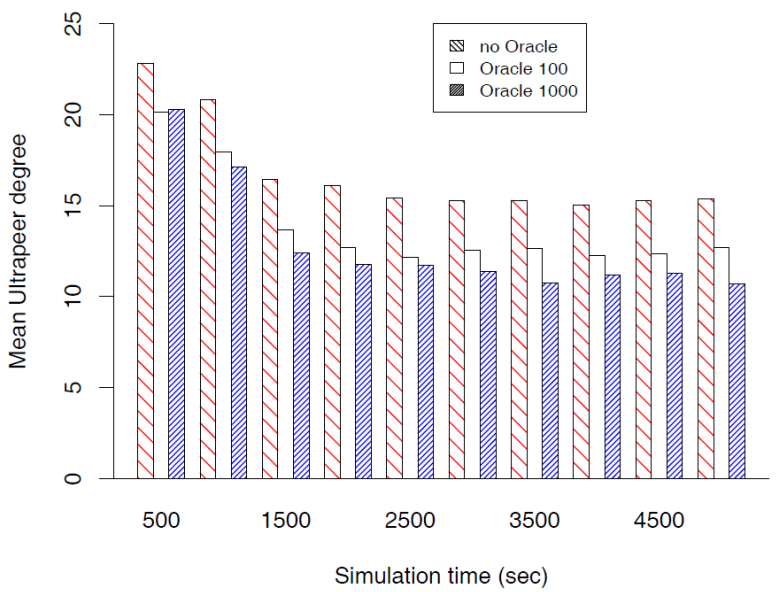


Figure 4.3: Average node degree of Ultrapeers

For a Gnutella overlay that is composed of ultrapeers and leaf-nodes, ultrapeers can, by design, maintain a much higher number of connections with other peers than leaf-nodes. However, in our analysis, we notice a similar node degree pattern among both types of nodes. We therefore report only on that of the more significant

ultrapeers. Figure 4.3 shows the average node degree of ultrapeers in each of the 3 cases. Despite a generally decreasing trend with time, for all 3 cases, the node degree decreases the most for the Oracle case with a list-size of 1000. Its largest difference of 4.54 units occurs at 3500 second into the simulation.

We also notice that the Oracle case with a list-size of 1000 started off having a slightly higher average node degree than that with a list-size of 100. However, with time, the latter shows increasingly better values than the former. Still, the average node degree remains generally high enough to not negatively impact the overlay topology.

Intra-AS Connections

We next look at the proportion of direct neighbor connections that ultrapeers build with other peers from within the same AS and compare it to their total number of connections.

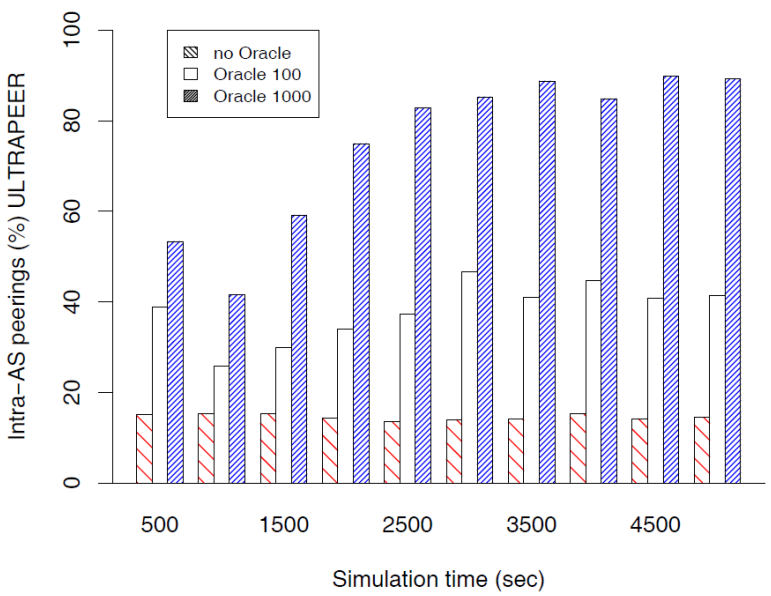


Figure 4.4: Percentage of Ultrapeer connections within the same AS

Figure 4.4 shows the percentage of intra-AS to total connections formed by ultrapeers. The proportion stagnates with time, for the unbiased case. Meanwhile, it increases for both cases of Oracle with 100 and 1000 file sizes respectively. The highest percentage difference of 74.95% to the unbiased case is recorded at 3500 seconds of simulation time by the Oracle with a file size of 1000. It also shows significantly better percentage values than those for the Oracle case with a file size of 100.

4.4.3 User Experience

Queries, Responses and Traffic Reduction

The Traffic in P2P overlay networks are of two major categories, signaling and data transfer. The signaling traffic consist mainly of connection negotiations and query searches/responses. In unstructured P2P overlay networks, the signaling traffic constitutes a considerable fraction of the total traffic.

	Queries			Responses		
TTL	Unbiased Overlay	Biased Overlay	% Change	Unbiased Overlay	Biased Overlay	% Change
8				5	152	2,940.00
7	19,725	11,149	-43.48	26	1,941	7,365.38
6	414,718	186,473	-55.04	363	11,284	3,008.54
5	3,611,604	986,261	-72.69	8,789	34,031	287.20
4	7,190,754	2,287,036	-68.19	67,381	58,488	-13.20
3	947,035	1,592,910	68.20	94,392	67,651	-28.33
2	30,653	497,464	1,522.89	97,305	69,003	-29.09
1	2,093	74,460	3,457.57	41	16	-60.98
0				22	10	-54.55
Total	**12,216,582**	**5,635,753**	**-53.87**	**268,324**	**242,576**	**-9.60**

Table 4.2: Number of queries and responses in P2P Overlay

Gnutella queries are forwarded with a default Time To Live (TTL) value of 7 from a source to all its directly connected neighbors. The TTL value is reduced by 1, each time the query gets forwarded by a successive neighboring peer. Forwarding stops when the TTL reaches 0. Consequently, because a query is propagated by means of flooding, a single query gets forwarded multiple times around the network, when searching for content.

We take account of the resulting query-related traffic and the number of responses that get routed back to the original source, for each query hit. Table 4.2 contains the recorded number of queries and responses. It shows that the highest number of queries are recorded at the same level of the propagation, i.e. when TTL=4, for the unbiased as well as for the biased case. The highest number of responses are also recorded at the same level, at TTL=2, for both cases. These confirm that the Oracle does not hinder queries from being propagated many hops away from the source nor from getting responses from peers that are much further away.

Despite the above similarities, we notice a drastic reduction in the query traffic from 12.2 million to 5.6 million packets, when the Oracle service is used. We as well notice that using the Oracle service causes a slight reduction in the number of responses, from 268.3 thousands to 242.6 thousands. Interestingly, for only a 9.6% reduction in the number of responses, more than half the amount of the heavier query traffic swarm (53.87%) is avoided. We consider this an acceptable trade-off. Such large reduction in traffic helps to free up crucially needed bandwidth and improve

network efficiency. Network efficiency benefits both the ISP and end-users. Particularly though, the traffic reduction without impacting application functionality and performance, is an added benefit to the ISP by the Oracle service.

Overlay Path Length

As can be seen in Figure 4.5, the average path length between overlay peers remains virtually unchanged (or changes only slightly) with time, for the unbiased and 100 list-size Oracle cases. Concurrently, a clear increase with time is recorded for the 1000 list-size Oracle case.

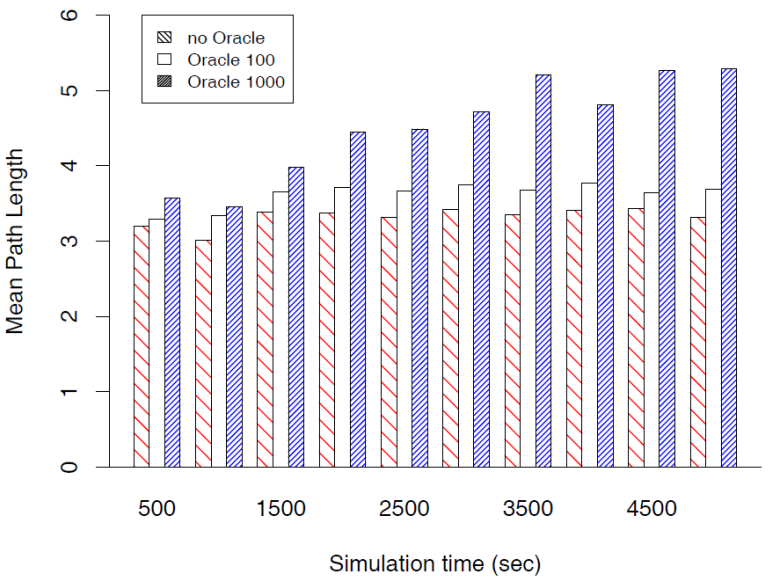


Figure 4.5: Average overlay path length

In spite of this slight increase in mean overlay path length, since more intra-AS connections and exchanges are now occurring within the same AS, when using the Oracle with a list-size of 1000, its final impacts are neutralized and show no effect on the performance.

Average Underlay AS Distance of Overlay Peers

The AS distance simply refers to the number of domains (or ASes) separating any two peers. This is easily determined by mapping each overlay peer to its underlay AS and then counting the number AS hops between them.

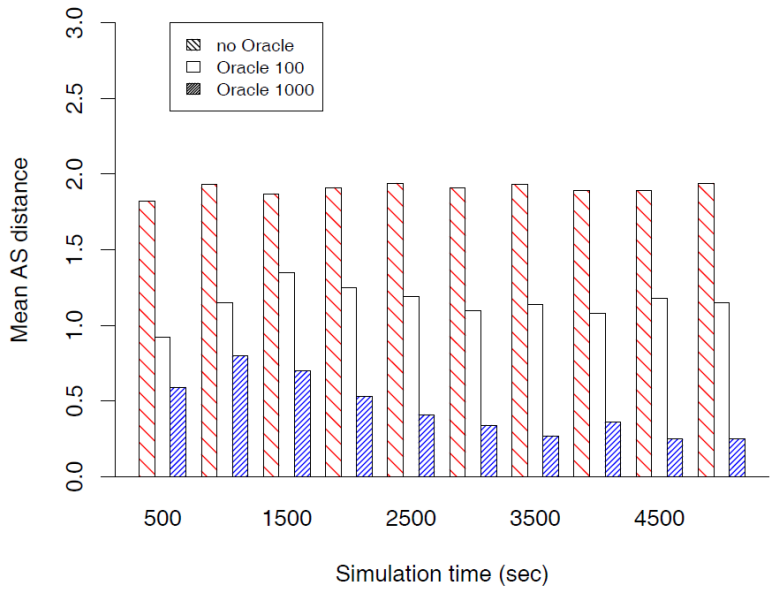


Figure 4.6: Average underlay AS distance of Overlay peers

Contrary to the trend we observed for the average overlay path lengths in the previous sub-section, Figure 4.6 shows a reduction in the average underlay AS distance with time, for the 1000 list-size biased case. The maximum reduction occurs at 5000 seconds, from a value of 1.94 to a value of only 0.25.

The very low average AS distance for the 1000 list-size biased case is simply an indication and confirmation of increased locality because of the Oracle service, as already seen in Section 4.4.2. For an ISP, this means retaining much more traffic within its own AS, which is a major cost saving objective.

4.5 Effects of Topology and User Behavior on Locality

To improve on the work done in the previous section, we design our networks to include different topologies and more realistic distributions for user behavior. We also include various user distributions, including their access bandwidths as a further criteria for the Oracle service to use in sorting the list of potential candidates.

We run two main sets of experiments. One set to analyze topological effects and the other set to analyze user behavioral effects. Each experiment is ran for two distinct cases; one unbiased (U) case, where the Oracle service is not used at all and one biased (B) case, where the Oracle service is first used during bootstrapping and later for content downloads. With these experiments, we seek to examine how

the Oracle's locality service is affected by different topologies and different user behavioral patterns, including adverse churn and content rarity.

4.5.1 Performance Metrics

To evaluate the experience of both the ISP and P2P user, we consider the following metrics:

- Number of responses generated per query
- Overlay hop count of the responses
- Underlay AS distance of the responses
- Download response time
- Proportion of exchanged content retained within an AS
- Quantity of reduced overlay traffic

All files have a standard size of 512KB, the same as the piece size used in most popular P2P systems. Peers connect with each other using TCP and use HTTP to exchange data. For each unsorted list the Oracle receives from peers using the service, it sorts it as follows:

i) First, identify peers in the same AS as the requesting peer, sort them by their bandwidth and place them at the top of the sorted list.

i) Then, use AS-distance to sort the rest of the peers, i.e. peers not in the same AS as the requesting peer.

We use the more realistic Weibull distribution to model online session lengths and content availability. Queries are propagated via flooding. The results in each of the experiments are based on 10.000 successful query requests that result in 10.000 successful downloads.

4.6 Evaluating Topological Diversity

To study how P2P locality is affected by different ISP underlay topologies and the distribution of peers within them, we design 5 different AS topologies, comprising 2 national and 3 world topologies. Each topology contains 700 P2P hosts, distributed as shown below in Table 4.3.

4.6.1 Designing the Topologies

The 5 AS topologies include:

- **Germany:** We retrieved a copy of Germany ISP topology map from [113] and extracted the 12 biggest ISPs from it, including their inter-AS connections. Using the broadband user information in [50], we distribute the 700 P2P hosts among the 12 ISPs, according to their fraction of broadband customers.
- **USA:** We model one regional provider per city for each of the 25 major cities and connect them using information obtained from [132, 183]. Each AS (city) gets a fraction of the 700 P2P hosts corresponding to its share of the population.
- **World1, World2, World3:** We model 3 different World topologies, each with a single Tier-1 AS, 5 Tier-2 ASes and 10 Tier-3 ASes, for a total of 16 ASes per topology. Interconnections between the ASes are designed according to routing information contained in [146].

	Tier1 (# AS / peers per AS)	Tier2 (# AS / peers per AS)	Tier3 (# AS / peers per AS)
World1	1 / 10	5 / 46	10 / 46
World2	1 / 355	5 / 23	10 / 23
World3	1 / 50	5 / 46	10 / 42

Table 4.3: AS and Peer distributions in the 3 World topologies

The 700 peers are distributed based on results from [131, 146]. Table 4.3 summarizes the different ways ASes and peers are distributed in the World topologies.

The above designs give us the possibility to study how different topologies and peer distributions affect the overlay/underlay performances.

We thus model these topologies within the SSFNet environment, taking the memory limitations and difficulties/constraints involved with simulating such large and complex networks within such an environment, into consideration [63].

Each AS has two routers with separate functions. One peering (inter-AS) router for connections with other ASes and one user-access (intra-AS) router for local connections with peers. Peers are connected in a star topology with this router. The peer connection speeds reflect the normal DSL/cabel modem speeds at the time. We use speeds ranging between 1 and 16 Mbps for this. Tier-1 and Tier-2 ASes contain larger proportions of higher speed subscribers [50, 146, 176], so we assign speeds of 10 - 16 Mbps to 80% peers in Tier-1 AS and speeds of 1 - 4 Mbps to 60% of peers in Tier-3 ASes. We also assign link delays between 4 - 6 msecs to

connections between Tier-1 and Tier-2 ASes and link delays between 18 - 20 msecs to connections between Tier-2 and Tier-3 ASes [132, 221].

4.6.2 Modeling User Behavior

Studies show similar user behavioral patterns across structured and unstructured P2P systems [86, 89, 90], despite continuous transitions. However, there are some differences between file sharing and video streaming P2P systems [92, 216]. Churn is one of the most well-studied and analyzed user behavioral character [190, 212]. We use different distributions to simulate observed behavior, as well as abstract and worse case scenarios. We employ *sensitivity analysis* to explore and determine parameters that best fit the distributions representing observed behaviors, i.e. within the limitations of accuracy possible in our simulation environment.

Shared Content

Content replication helps improve download speeds and general performance in P2P networks. However, a significant number of users are selfish and share nothing at all (free-riders). This greatly impacts the overall availability of content. *Free-riding* is confirmed by several P2P measurement and analysis studies [1, 94, 128, 168, 176]. In conclusion, they stipulate that the number of files shared by peers approximates a heavy-tail distribution. We thus use different distributions to model shared content in the overlay network.

The number of files that each peer shares is plotted against the peer's ID as shown in Figure 4.7, using the following distributions:

- **Uniform distribution** with parameters, min=0 and max=100, to represent the comparison baseline (Figure 4.7a).
- **Pareto distribution** with parameters, k=100 and alpha= 10, as one form of a long-tail distribution with majority peers sharing relatively moderate to small number of files (Figure 4.7b).
- **Weibull distribution** with parameters, scale=4.2 and shape=0.5, to represent another form of long-tail distribution with few peers sharing a large number of files, while a good number of peers are free-riding and sharing zero files (Figure 4.7c).
- **Poisson Distribution** with a mean of 50, representing the hypothetical case that a constant number of files are shared at any given time (Figure 4.7d).

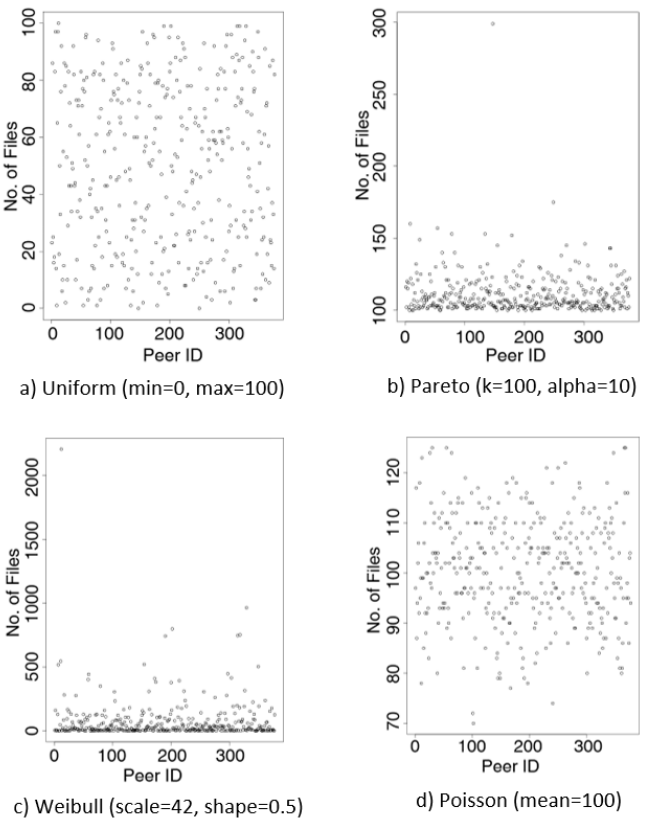


Figure 4.7: P2P content distributions

In real P2P networks, the types of content that are shared, often depend on their popularity. Less popular contents are often difficult to find. Most users stop sharing less popular and outdated contents to create/retain space for more recent and popular ones.

Session Length

Two important characteristics of peers in overlay networks, are the randomness of their participation and the length of time they stay online. The dynamics of this participation is what is generally referred to as *churn*. Churn affects session length, which can also be influenced/limited by the ISP, e.g. through enforcement of 24 hours session timeouts [137].

Extensive studies have been done on churn and online session lengths [86, 190, 203], with conclusions that the latter fits a heavy tail distribution with varying parameters. A good understanding of churn is necessary to effectively design and model P2P systems. To this end, we again use four representative distributions

to model the concurrent length of time that peers stay online without quiting or bouncing (i.e. session length).

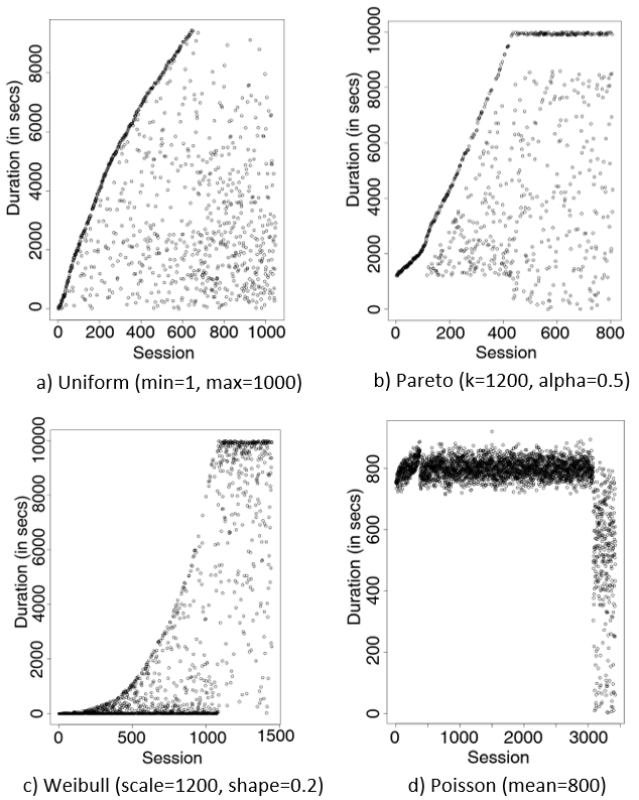


Figure 4.8: Session length distributions

Session length also affects content availability. Contents can only be successfully shared, if the peers that are sharing (uploading/seeding) stay online long enough to allow others download and seed as well.

Although downloads can also be completed after multiple sessions, it often happens that the original seeder of a particular content, starts sharing, but goes offline and does not return for a long period, or pulls back the content for good, before other peers have the opportunity to complete downloading it. Such incomplete dowmloads contribute to the fraction of shared content classified as junk.

Queries

There are two major types of queries, i.e. *constant phrases* that search for *specific types of content*, e.g. mp4, mp3, ebooks, and *volatile phrases* that search for *specific*

contents, e.g. names/titles of authors/artists/albums/books. To enable us study the effect of P2P locality on searches, we model our queries to represent 45% of each type, mimicking their popularity distributions and loads, as reported in [74, 121]. We then model the remaining 10% to match no available content. In modeling the queries, we ensure that they are modeled in a way, which ensures that 20% of all queries result in at least 1 or at most 2 hits.

4.6.3 Simulation Results and Analyses

The results of the first set of experiments involving different topologies (and their impacts) are presented in the sub-sections that follow.

Structural Properties of the Overlay Topology

Since the Oracle service is first used at bootstrapping to effect the construction of a more localized overlay, we start with investigating the structural properties of the Oracle-biased overlays and compare them with those of the unbiased overlays.

Unbiased overlays are characterized by the following graph structural properties:

- remain connected (no disintegration into subgraphs) despite churn
- possess small graph diameters
- low average path lengths
- low average node degrees

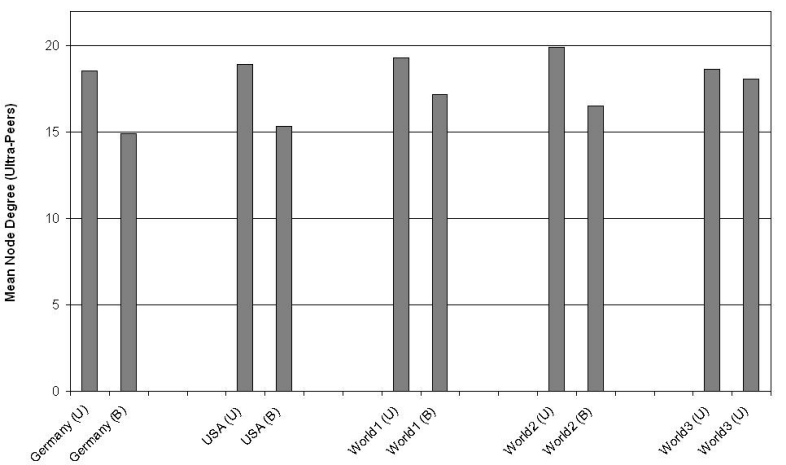


Figure 4.9: Average node degree of overlay peers

In comparison with Oracle-biased overlays, we observe that these too remain connected in all but a few sampled instances. However, these instances are only temporal, since resulting subgraphs get joint again a few seconds later.

We also observe a slight change (decrease) in the average node degree of Gnutella ultrapeers in the biased overlay. This can be seen in Figure 4.9. It shows comparatively small differences, of less than 3 units to those of ultrapeers in the unbiased overlay. USA and Germany are extreme cases portraying the biggest differences.

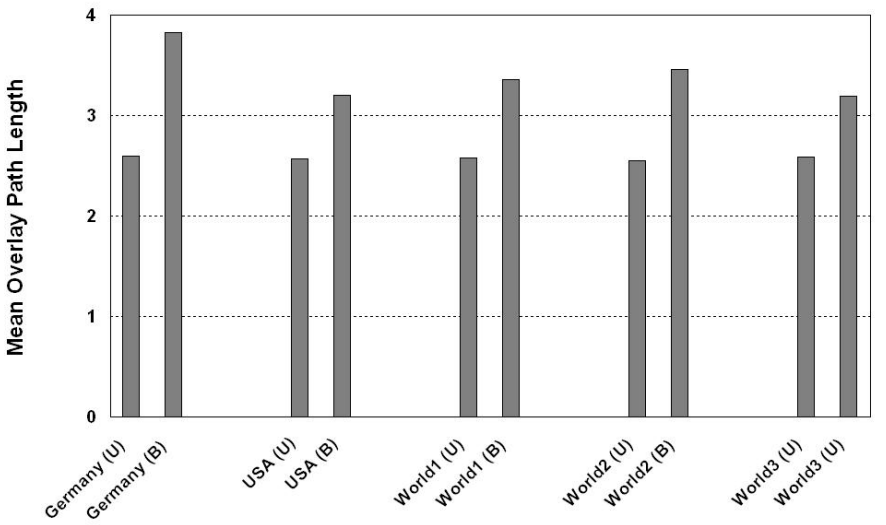


Figure 4.10: Mean Overlay Path Length

Figure 4.10 shows the average overlay path length of the unbiased and biased topologies. We notice that although it slightly increases in all 5 biased topologies, the difference remains well below 2 magnitude points. In fact, in the extreme case of Germany topology, where the difference is highest, it is only slightly above 1 magnitude point.

Queries/Responses Analysis

The effect of locality on the number of responses is shown in Figure 4.11. It shows an increase in the number of responses for the biased case, across all 5 topologies. With increased locality via use of the Oracle service, more queries get sent to proximal neighbors within the same AS, which also means much more responses are coming from local neighbors within the same AS.

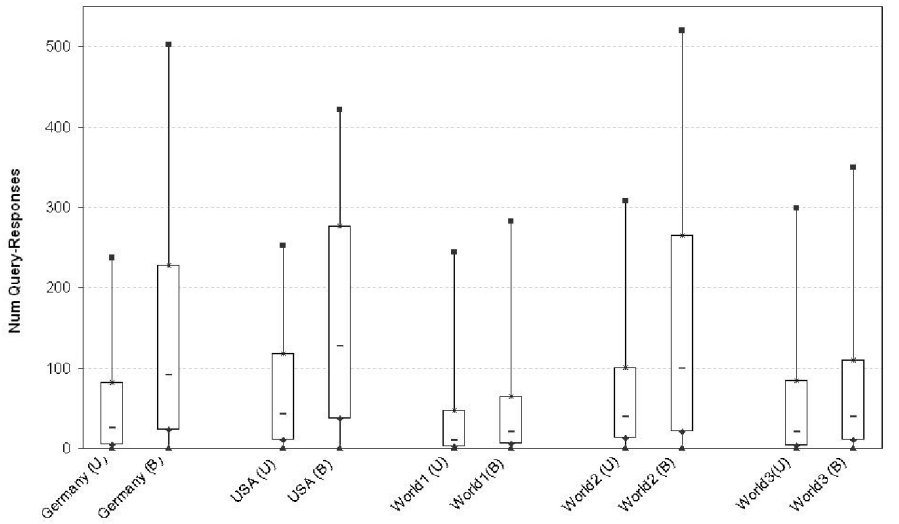


Figure 4.11: Number of Responses per Query

Since queries are propagated via flooding, this also means less signaling traffic is being flooded beyond an AS's boundaries. It eventually reduces the volume of inter-AS traffic and increases the proportion of intra-AS traffic.

Content Downloads

Understandably, users join a content (file) sharing P2P overlay to share (download and upload) content. For most users though, sharing simply means *always downloading and never uploading*. Their sole purpose is *download at all costs and upload at no cost*. This is evident in the high levels of selfishness and free-riding observed in P2P networks. With downloads having such high significance, we use *download response time* as the appropriate metric to quantify the end user's experience.

We see in the box plot of Figure 4.12 that the average time taken to download a 512KB file reduces across all 5 topologies, by 1 - 3 seconds in favor of the Oracle biased case. This is equivalent to a reduction of 16 - 34% in download times.

Download speeds generally also depend on the size of a candidate peer's last mile bandwidth. Therefore, a peer in a different AS might be a better candidate to download from than one in the same AS, if its upstream bandwidth is higher.

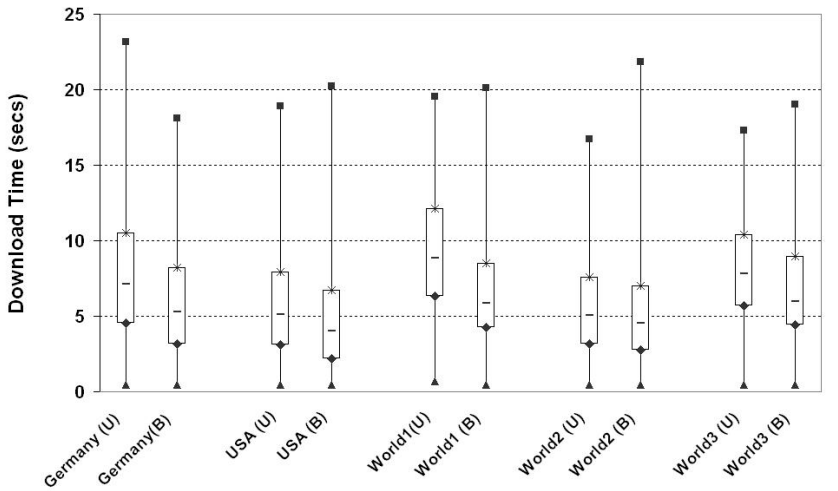


Figure 4.12: Download response times

Intra-AS Exchanges

We saw in Section 4.4 that using the Oracle service causes more peers within the same AS to form more localized connections with each other. As a result, the proportion of intra-AS exchanges also increases. We repeat the same procedure on the 5 topologies being investigated in this section.

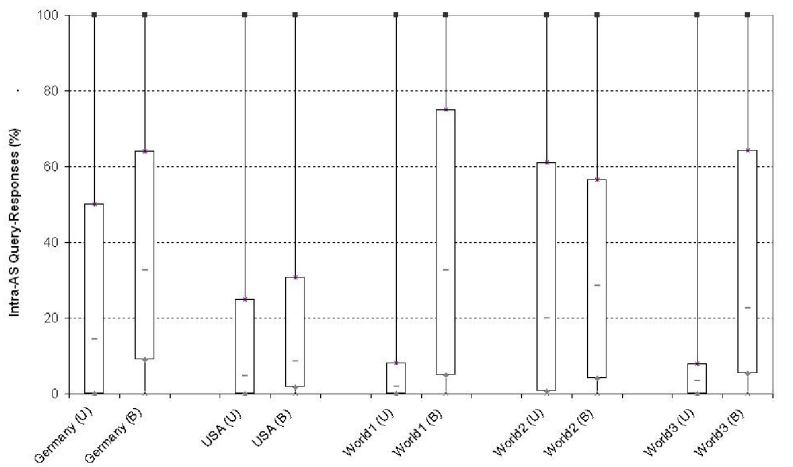


Figure 4.13: Percentage of responses from within the same AS

For topologies using the Oracle service, the box plots in Figure 4.13 show marked increases in the average proportion of responses coming from peers that are within the same AS domain as the peer that sent the original query. The increase is consistent across all 5 topologies.

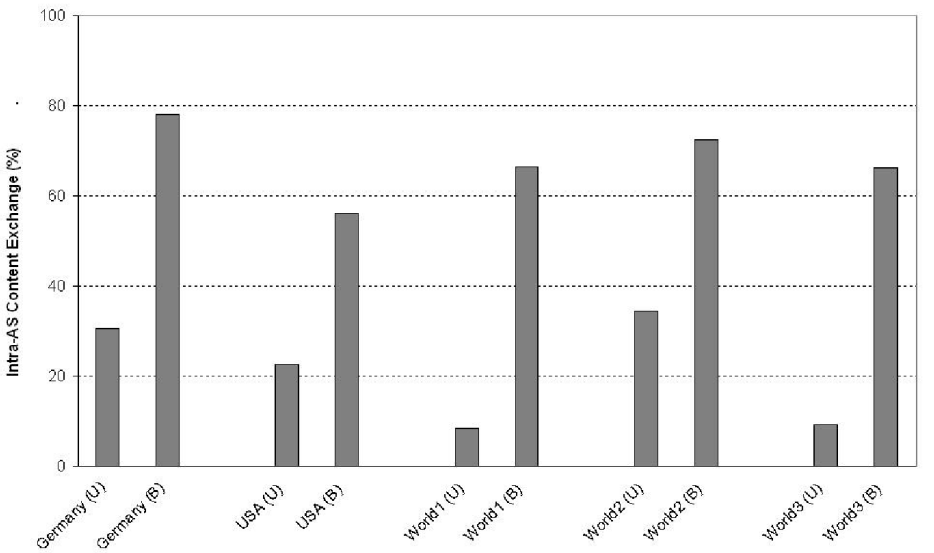


Figure 4.14: Percentage of content exchanges between Peers in the same AS

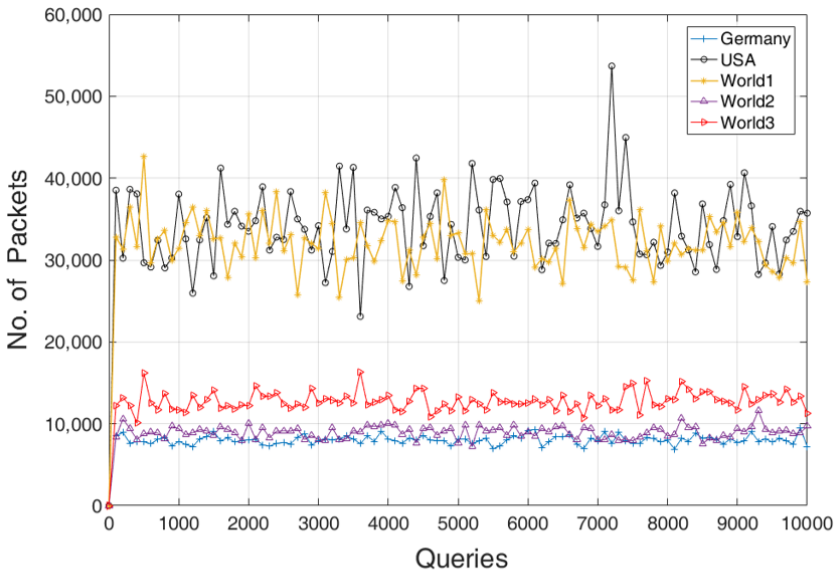
In tune with the increased proportion of responses observed coming from peers within the same AS as the peer that sent the query, we also observe a corresponding increased proportion of content exchanges within the AS, when the Oracle service is used. Figure 4.14 shows that for each topology, the percentage of intra-AS exchanges is much higher for the biased case than for the unbiased case.

Inter-AS Traffic Reduction

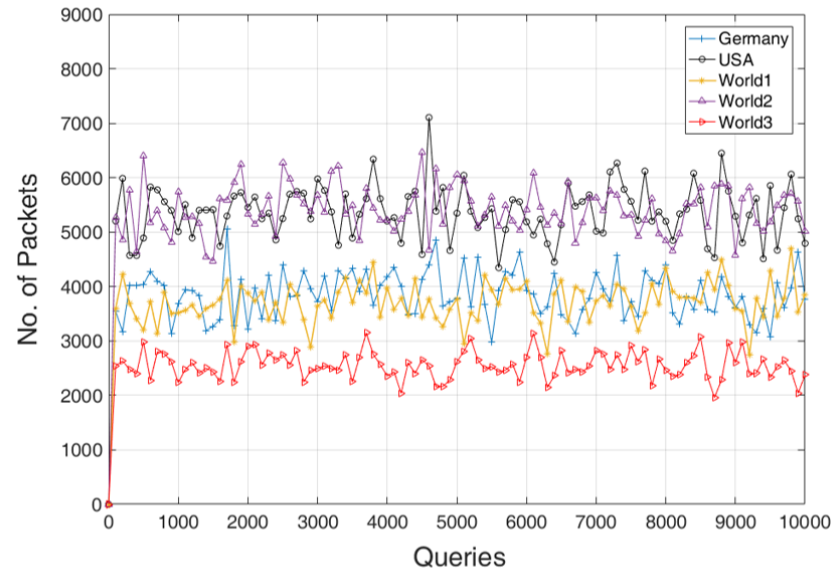
With more responses to sent queries and content exchanges between peers, now occurring within an AS domain, as a result of the Oracle locality service, we proceed to analyze the change in the amount of signaling traffic that is exchanged between ASes.

Figure 4.15 shows the average number of packets that are exchanged per query between ASes in each of the 5 topologies. In comparing the unbiased case, Figure 4.15a, where the Oracle service is not used, against the biased case, Figure 4.15b, where the Oracle service is used, we notice a huge traffic reduction across all biased topologies using the Oracle service. World1 and World3 have close to 6-fold and 10-fold traffic reductions, respectively.

This is a further confirmation of the Oracle service's traffic management abilities and an added benefit to the ISP. Eliminating or at least, containing a good portion of such traffic swarms within an ISP's own AS, relieves bandwidth consumption on peering and transit links. Thus, a further ISP costs-saving aspect of the Oracle solution.



a) Inter-AS signaling traffic (without Oracle)



b) Inter-AS signaling traffic (using Oracle)

Figure 4.15: Traffic reduction across domains

4.7 Evaluating Changes in User Behavior

Using a mix of national and world topologies, we've shown that both the ISP and the user do benefit from increased locality via the Oracle service. The benefit remains consistent across different topologies. In this section, we'll now investigate if the same holds true for different (extreme) user behavioral patterns.

We therefore extend the user behavioral patterns for *session length* and *shared content* by including Uniform, Pareto and Poisson distributions to the Weibull distribution already used in the previous section. The total of 16 different combinations, resulting from the 4 shared content and 4 session length distributions, offer us the possibility to also study such extreme conditions, as adverse churn in the presence of sparsely available content.

Since our emphasis in this section is more on user behavior than on topology, to minimize the effects of topology, we select the one that has the most evenly distributed peers in all its ASes. We thus select the World3 topology to run the 16 experiments on.

We then analyze the results based on the metrics outlined in Section 4.5.1.

4.7.1 Average Node Degree and Path Length of Overlay Peers

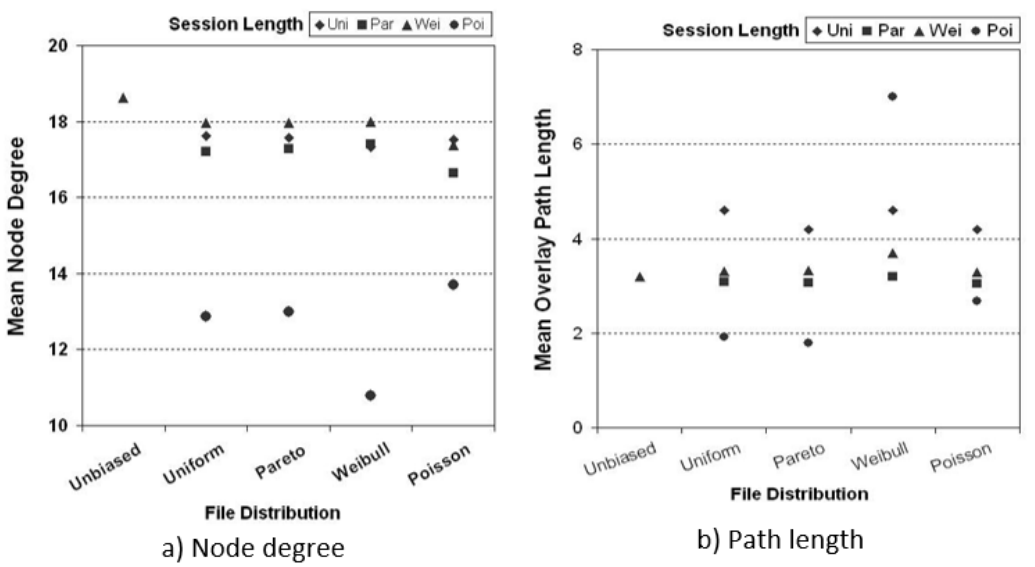


Figure 4.16: Average node degree and path length of overlay peers in World3 Topology

The average node degree in the unbiased case is only slightly higher than those of the other distributions, as can be seen in Figure 4.16a. The only exception is with the Poisson session length, which has a difference of between 4 and 8 average node degrees to that of the unbiased case. It also has the least average node degree in combination with each of the 4 shared content distributions.

Figure 4.16b shows that the average overlay path length of the biased overlays (those based on the given distributions) practically fall within the same range as that of the unbiased overlay. The only noticeable exception in the biased case occurs when the session length is Poisson and the file distribution is Weibull. This is when the highest mean overlay path length of approximately 7 is recorded.

4.7.2 Queries/Responses Analyses

We analyze the responses that other peers generate and send back to the peer that sent the original query request.

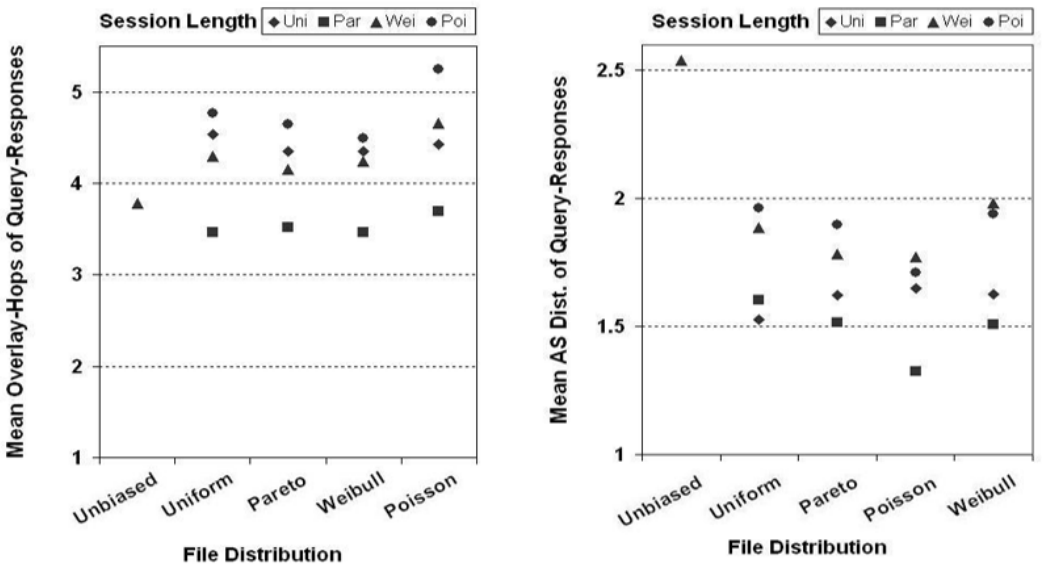


Figure 4.17: Average overlay hop count and underlay AS distance of responses in World3 Topology

Although Figure 4.17a shows that the average overlay hop count of these responses are slightly higher for most of the biased combinations than it is in the unbiased overlay, Figure 4.17b shows the contrary, with regards to average AS distance of the underlay. This simply confirms increased locality as a result of using the Oracle service.

4.7.3 Intra-AS Content Exchanges and Download Times

We observe much higher percentages of intra-AS content exchanges across all 16 combinations of the Oracle's user behavioral patterns than in the unbiased case that does not use the Oracle service. Figure 4.18a shows a difference of at least 48% between the least performing Oracle pattern (Poisson session length and Poisson file distribution) and the unbiased case.

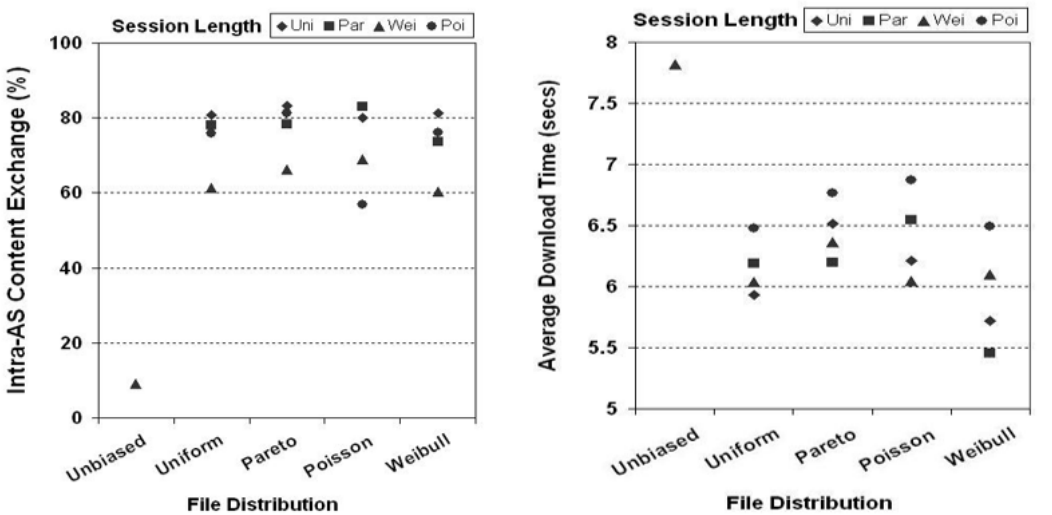


Figure 4.18: Intra-AS Content Exchanges and download times in World3 Topology

Correspondingly, Figure 4.18b also shows that the average download times of all Oracle-based distributions are much better (lower) than that of the unbiased case. The least performing Oracle case, with the least difference (highest average download time) compared to the unbiased case, occurs with Poisson session length and Poisson file distribution.

4.8 Beyond the Oracle Service

Operating an efficient multi-purpose network, such as the Internet, requires careful planning and intelligent allocation of often limited resources. ISPs face mounting challenges in keeping up with the growing proportions of traffic volumes and flows, over which they have only limited or no control. So far, challenges caused by P2P can effectively be addressed via collaboration enabled and promoted by the Oracle service.

In terms of complying with security, privacy and protection regulations, such as the General Data Protection Regulation (GDPR) [209] of the EU, the Oracle service can ensure that no information received from a requesting peer is saved or even cached. Further, instead of sorting lists based on specific IP addresses, a more generalized form based on network prefixes, e.g., /24 prefixes, could be used. To protect against Distributed Denial of Service (DDoS) attacks when requests originate from spoofed IP addresses, the Oracle can limit the number of replies that are sent to a specific IP address. Additionally, since the size of the response sent back by the Oracle to a requesting peer, is the same as that of the request that the peer sends, it can not be used for amplification attacks.

The monstrous demand and consumption of high quality media contents has overtaken the use of P2P and is causing a major shift from P2P to CDNs. The Oracle service offers huge potentials that go beyond the scope of P2P. It can easily be adapted to suit collaborations with CDNs as well.

On the one hand, CDNs want to deliver contents to end-users in the fastest and most efficient way possible, but sit at the source and have no control over the path that requested contents take to reach end-users. On the other hand, ISPs provide the paths used to deliver these contents, but lack control over their sources. A lot is at stake for all parties involved, i.e. the CDNs, the ISPs and the end-users. The Oracle services provides the fundamental building block for a suitably adapted solution in this case, as well. In fact, researchers have adapted and extended the principles of the Oracle to enable ISP-CDN collaboration [72, 161] as well as to enable Content-aware Traffic Engineering for traffic originated by CDNs [71]. This has even evolved into finished products (BENOCS Director [76] and BENOCS Analytics [75]) and has resulted in the creation of a business Start-up, which is in-charge of further developing and commercializing the products. Some global Tier1 ISPs and CDNs are already on-board.

The Oracle service is also a major contributor to the standardized ALTO protocol, created through an initiative of the IETF ALTO Working Group. The Internet draft "The PROXIDOR Service" [7] based on the Oracle Service is integrated in the final ALTO protocol [10] in RFC7285.

4.9 Summary

It is no doubt that new trends and phenomena will continue to emerge on the Internet. It is also no doubt that, when the proportion of traffic they generate becomes quite significant, network operators and researchers will become interested. Their effects on ISP topologies and traffic management approaches will warrant detailed studies and (if need be) appropriate mitigation approaches. Recently, P2P has warranted appropriate approaches to deal with the effects of its enormous traffic proportions.

In this thesis and in this chapter, we outlined some of the issues posed by P2P systems and P2P traffic. We therefore proposed a solution that we think should address them to the benefits of both the ISP and the P2P end-users. Our proposal is based on the use of a simple ISP-operated value-added service, the *Oracle* service, which helps peers make informed and better decisions.

To assess the benefits of the Oracle service, we implement the Gnutella P2P protocol in a packet-level simulator, then design and model AS/P2P topologies, with which various aspects of the P2P and ISP collaboration are studied. To demonstrate the advantages of using the Oracle service, we perform comparative simulation studies and analyze their results. We use a visualization technique to show that the overlay topology becomes more aligned with the underlay topology, when the Oracle service is used and also that the overlay graph remains connected. Despite slight increases in average overlay graph diameter and slight decreases in the average node degree, we still record a substantial increase in intra-AS connections in the Oracle-biased topologies.

The user experience also improves, as evident in the recorded average number of query responses and the average download times, which are significantly better for the Oracle-biased cases. We also observe huge reductions in inter-AS overhead traffic. In some topologies, the biased inter-AS traffic is reduced to as little as one-sixth and one-tenth of what they are in the unbiased case, which is a 6-fold and 10-fold reduction, respectively.

5

Traffic Effects on Different Backbone Topologies

We perform comparative studies in this chapter through evaluation of network and application performances in three different backbone topologies under high traffic load. All three topologies have the same number of nodes, but differ from each other by the number of links (i.e. 66, 30 and 20 respectively) connecting these nodes. They also differ with respect to the total capacity in the topology and in the nature of their interconnections. We also study their respective responses to an increased traffic load of 35%, as well as, to a single link failure, i.e. when the link with the highest throughput in each topology fails.

New trends and changes in user behavior can affect traffic diversity, volumes and flows in various ways. Network Providers adapt by re-engineering their infrastructures to accommodate the effects that accompany such trends and changes. To study how the topology of backbone networks affect traffic flows and overall network/application performance, we use a reference backbone network model for Germany [17, 93, 95]. We modify the number of nodes to 12 (in accordance with a more recent reference model (IDEALIST project [39]), then vary the number of links and the nodes they interconnect, to obtain 3 dissimilar topologies. We then apply similar loads to each topology and analyze their performances based on various criteria.

5.1 Backbone Topologies

Topology design is an essential part of the traffic management solution. Long-haul backbone transport networks, which are generally characterized by very high construction and operational costs, are also mostly static in nature. Once designed and built, they maintain the same structure for decades, despite latest conditions that necessitate a more flexible topological structure, which can easily and flexibly contain the growing and increasingly more dynamic traffic volumes and flows. ISPs instead turn to invest more on faster and bigger routers/links than in redesigning the architecture of the underlying transport network. Easier topological changes, such as links addition/removal/moves are however more probable and often preferred.

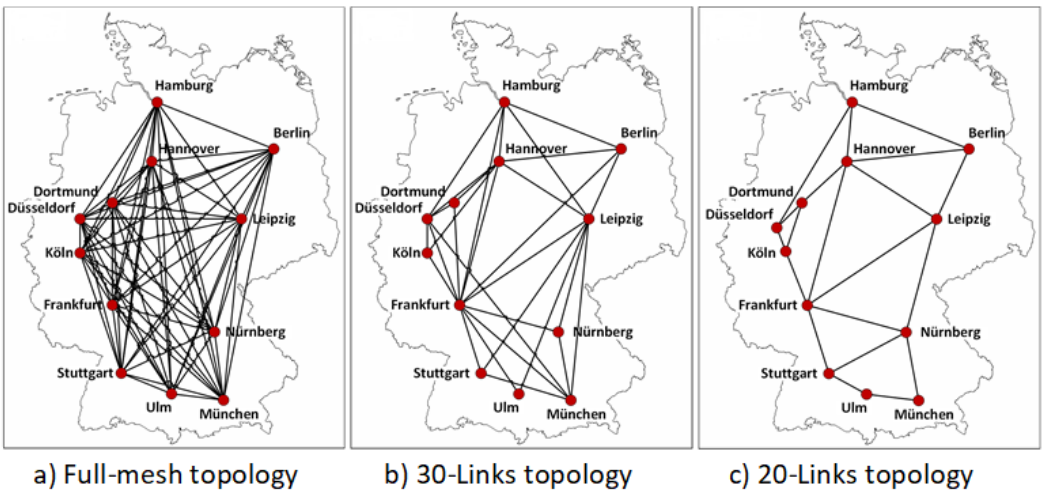


Figure 5.1: Germany Backbone Topologies

Topology	# Links	OC-192 (10Gb/s)	OC-48 (2.5Gb/s)	Total BW (Gb)
Full-mesh	66	16	50	285
30-Links	30	16	14	195
20-Links	20	11	9	132.5

Table 5.1: Summary of Backbone Topologies

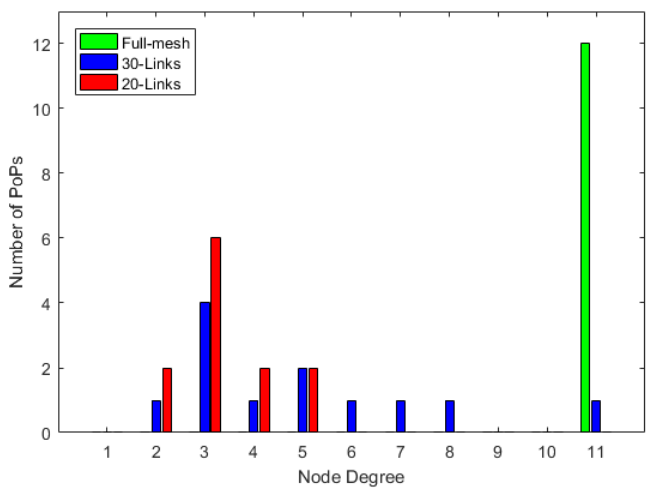


Figure 5.2: Node Degree Distribution

For our studies, we model one full-mesh and two partial-mesh topologies. The nodes represent Points of Presence (PoP) or major Data Center (DC) locations. The interconnections between the nodes represent national backbone links. The interconnections and node degree distribution vary from topology to topology. Figure 5.2 shows the node degree distributions in each topology. The higher the node-degree, the better connected that node is. Each topology also differs from the others in terms of their total bandwidth, as can be seen in Table 5.1.

5.1.1 The Fullmesh Topology

All PoPs in the first topology are directly interconnected with each other, constituting a full-mesh. This is also the reason why we've named it the *Fullmesh* topology. The Fullmesh topology thus comprises 12 PoPs and 66 links. Each PoP has a node-degree of 11, which is the average node degree for this topology, as well as the highest possible node degree for this and the other 2 topologies.

5.1.2 The 30-Links Topology

The 30-Links topology forms a partial-mesh that connects the 12 PoPs using 30 backbone links. A little less than half as much links as are present in the Fullmesh topology. A unique feature of this topology is that, one PoP (Frankfurt), from where the largest volume of traffic to other PoPs is sourced, also connects via direct backbone links to each of the other 11 PoPs. Thus, the node degree of the Frankfurt PoP is 11, while that of the other PoPs vary between 2 and 8. The average node degree in the topology as a whole is 5.

5.1.3 The 20-Links Topology

The 20-Links topology also forms a partial-mesh. However, it has the least number of backbone links between its 12 PoPs and no unique feature, such as direct connections between top talking PoPs. The node degree in this topology varies between 2 and 5, resulting in an average node degree of only 3.3, which is also the lowest average value of all three topologies.

5.2 IP Traffic Demand Matrix

The IP traffic demand matrix represents the volume of IP traffic that flows between each Origin-Destination (OD) pair in the topology. We use a traffic demand matrix, which represents peak hour load captured at 15-minutes interval. Demand matrices

that are captured at shorter time intervals, e.g. of 10 or 5-minutes, are more accurate and more sensitive to load variations. However, we assume minimal OD load variation during the peak hour and therefore consider the 15-minutes interval to be ok for our purposes.

5.3 Simulation Studies

As mentioned in section 5.1, the topology of a network is an important design element with far reaching implications on its performance. To evaluate these effects, we design and run performance-based experiments on each of the three representative topologies described above. We then use selected metrics to analyze and compare their performances under the same network conditions.

5.3.1 The OPNET Modeler Simulator

Experiments on (or involving) backbone networks are rarely possible on real networks because of the high risk of impacting production and the serious implications that could follow from that. Researchers and operators resort to simulations for such studies. An appropriate simulator needs to be chosen, one that perfectly meets the conditions and requirements of the system to be studied.

After evaluating some of the most popular network simulation tools, we decided to settle with the Optimized Network Engineering Tool (OPNET) [111]. OPNET Modeler[1] is a packet-level event-based network modeling, simulation and analysis tool that provides many advantages that most of the other tools do not offer. These include its comparably very high simulation speed, the possibility to simulate very large communication networks using a detailed library of editable models that support existing protocols, an extensive list of multi-vendor modules and libraries, as well as interfaces for plug-ins. It also allows the design and study of networks, devices, protocols and applications with great flexibility. It is widely used in the research community and by network operators for planning, analysis and performance evaluations [178].

5.3.2 System Design

Each topology comprises 12 PoPs, representing 12 major German cities, and a predetermined number of backbone links. The number and size of the interconnections within each topology are given in Table 5.1. These bandwidths are modeled to suit the simulation-environment, such that 2.5Gbps simulated bandwidth represents

[1] OPNET Modeler is now called Riverbed Modeler [198]

10Gbps in real and 10Gbps simulated bandwidth represents 40Gbps in real, respectively. The link delays are based on measured averages or calculated from fiber lengths [32, 184].

The number of users (or clients) is the same (180) in all 3 topologies. These clients are distributed among the 12 PoPs, in proportion to their respective traffic demands and as shown in Table 5.2. They serve as originators of the different requests for applications/services available via connections to remote DCs. Only connections and traffic to/from remote DCs are considered and implemented because of their relevance in our studies. Local connections and traffic to/from local servers, i.e. those that do not cross a backbone link, are completely left out or ignored in our analyses.

All application servers in the DC are designed to be large enough, in terms of processing power and speed, to enable them respond to all client-requests in an effective and timely manner. This is necessary, to avoid unwanted bottlenecks that could impact the results in our studies. The type and number of application servers is uniform across all DCs (PoPs).

The backbone router is uniform across all PoPs. Open Shortest Path First (OSPF) is used as the IGP routing protocol of choice [2]. OSPF's advanced features, such as better (topology-dependent) metrics and equal-cost load-balancing also play critical roles in our studies.

5.3.3 Traffic Model

Two types of traffic are realized in the system. The first type is derived from a traffic demand matrix, a necessary input for such studies, and serves as the background traffic between all OD PoP pairs. The second type of traffic comes from modeled applications, possessing packet-level details and generated as a result of interactions between the clients and servers [15, 43, 85, 178].

The same background and application traffic volumes and flows (traffic demand between all SD pairs) are used in each of the three topologies. However, the path and hence the performance of each flow, which is a function of the topology through which it flows, varies according to the properties of that particular topology.

5.4 Network Performance Analyses

The used traffic demand matrix represents the average of that measured during the peak (or busy) hour. To emulate this peak hour condition, we also run all

[2] Although, we decided to stick with OSPF, it should be noted that initial experiments were also done with Intermediate System - Intermediate System (IS-IS), another link-state IGP similar to OSPF, with very similar results.

Location	Send (Gb/s)	Send (%)	Location	Receive (Gb/s)	Receive (%)	# Clients
Frankfurt (F)	22.61	54.84	München (M)	5.24	12.64	29
Düsseldorf (D)	6.94	16.84	Frankfurt (F)	4.92	11.87	24
München (M)	2.95	7.15	Stuttgart (S)	4.64	11.20	17
Hamburg (HH)	1.97	4.77	Hannover (H)	4.52	10.91	16
Berlin (B)	1.93	4.68	Nürnberg (N)	3.81	9.20	14
Leipzig (L)	1.09	2.64	Leigzig (L)	3.29	7.94	13
Stuttgart (S)	0.93	2.25	Hamburg (HH)	3.27	7.90	16
Hannover (H)	0.92	2.24	Dortmund (DO)	3.03	7.30	8
Nürnberg (N)	0.84	2.03	Berlin (B)	2.93	7.08	14
Köln (K)	0.36	0.87	Köln (K)	2.39	5.77	14
Dortmund (DO)	0.36	0.86	Düsseldorf (D)	2.34	5.64	9
Ulm (ULM)	0.34	0.83	Ulm (ULM)	1.06	2.56	6
Total	**41.23**	**100**		**41.45**	**100**	**180**

Table 5.2: Summarized Traffic Matrix (including clients distribution per location)

experiments for 3600 seconds. We ensure that all services and applications start only after routing convergence has occurred in the topology, which is typically within the first 10 seconds, but not later than the first 100 seconds.

Four important performance scenarios are studied:

- Performance under baseline traffic condition
- Performance under baseline traffic but in the presence of a single link failure
- Performnace when baseline traffic increases by 35%
- Performance when traffic increases by 35% plus a single link failure

Global peak hour Internet traffic is expected to grow 4.6-fold from 2016 to 2021 at a compound annual growth rate of 35% [108]. This is taken into consideration in our design and is the reason why we also select 35% as the rate for traffic growth in all three topologies.

For failure analyses, the link with (i) the highest throughput and (ii) the highest utilization in each of the 3 topologies is selected. There are various reasons for link failures, e.g. fiber cuts, port failure, router malfunction, etc. Our study does not dwell on these causes, but rather on their general effect, i.e. when the link (for whatever reason) suddenly becomes unavailable (or fails).

In their analytical study of link failures in an operational IP backbone network, Iannaccone et al observed that 10% of link failures last longer than 20 minutes, while 40% last between one and 20 minutes and 50% last less than one minute [97]. For failure analyses, we therefore consider a single link failure scenario in our models and select a realistic failure duration of 10 minutes during a typical peak hour.

5.4.1 Packets Hop Count

The hop count is the total number of IP layer devices (routers) that a packet goes through before reaching its final destination. We extrapolate the hop count of all successfully received packets, as shown in Table 5.3, then analyze them per topology and scenario.

Hop Count	3	4	5	6	7	8+
Baseline Traffic						
Fullmesh Topology	43,941,383	0	0	0	0	0
30-Links Topology	30,117,245	14,426,384	0	0	0	0
20-Links Topology	18,320,212	17,249,472	5,539,229	3,649,280	0	0
35% Traffic Increase						
Fullmesh Topology	44,898,812	0	0	0	0	0
30-Links Topology	31,117,135	13,532,826	0	0	0	0
20-Links Topology	16,493,492	17,595,538	5,512,175	2,513,505	42,594	0
Baseline Traffic + Link Failure						
Fullmesh Topology	43,451,616	451,633	0	0	0	0
30-Links Topology	29,589,247	14,833,325	100	0	0	0
20-Links Topology	17,927,592	16,545,115	5,826,114	3,488,364	295,400	0
35% Traffic Increase + Link Failure						
Fullmesh Topology	44,362,066	461,257	0	0	0	0
30-Links Topology	30,395,737	14,053,495	78	0	0	0
20-Links Topology	15,990,524	1,704,798	5,608,253	2,557,816	167,532	1,604

Table 5.3: Packets Hop count (Received traffic)

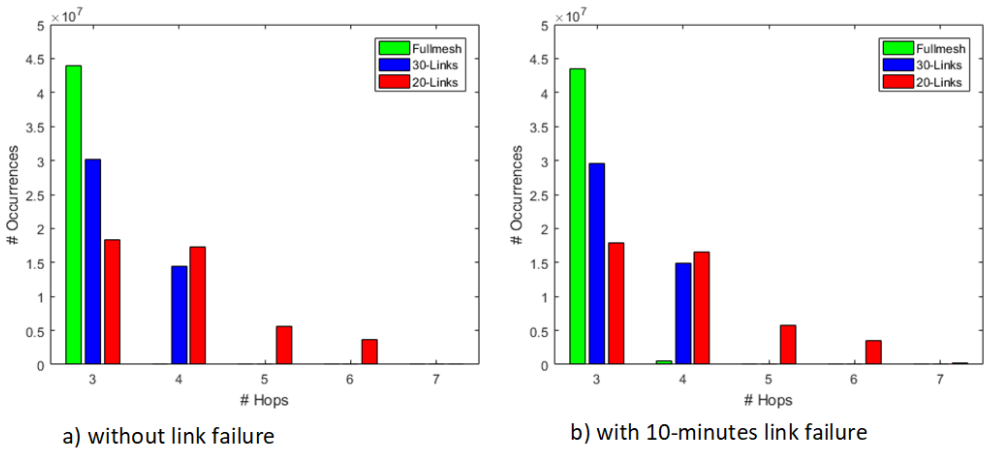


Figure 5.3: Average packet Hop Count at baseline traffic

As shown in Figure 5.3, packets in the Fullmesh topology travel the least distances to get to their final destinations. This is an attribute (and advantage) of a fullmesh

topology, since access to all other remote PoPs is possible via directly connected links. To go from a source to a final destination, packets that are exchanged between end-systems need to travel a single hop to their local gateway router, then a single hop across the backbone network and one more hop from the remote router to their ultimate destination. This equates to only 3 hops for any client/server remote communication in the Fullmesh topology.

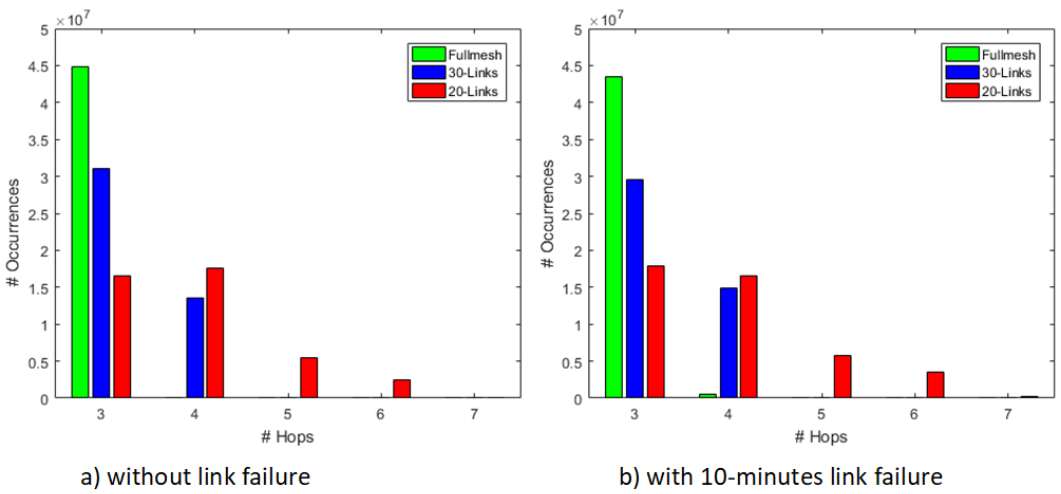


Figure 5.4: Average packet Hop Count at 35% increased traffic

5.4.2 Link Throughput and Utilization

The throughput of a link is a measure of the amount of data that is successfully transmitted via the link in bits/s. Link utilization is the ratio of its used bandwidth to its maximum bandwidth, expressed as a percentage. Since our topologies have links of different bandwidths (10Gbps and 2.5Gbps respectively) and since a major goal of any network is to push as much traffic through as possible, we use throughput instead of utilization as our measurement and comparison criteria. We therefore consider the performance of the top 10 links in each topology, with respect to throughput.

Fullmesh Topology

Under baseline traffic condition, the average throughput of the top 10 links in the Fullmesh topology ranges only between 1.25Gbps and 4.04Gbps, as shown in Figure 5.5a. This comparatively lower throughput is also accompanied by comparatively low link utilizations, resulting from the high diversity of direct and indirect paths of comparable bandwidths and metrics between any two PoPs in the topology. The

high volume of traffic is thus distributed among these links, such that, the average load and utilization on individual links remains low.

When the traffic in the topology increases by 35%, the average throughput also increases to a range between 1.68Gbps and 5.45Gbps respectively, as can be seen in Figure 5.5b. The same top 10 links are maintained, however, the ranking between the 8th-placed link (D - DO →) and the 9th-placed link (HH - D ←) in the baseline traffic scenario, is switched in the increased traffic scenario.

The Frankfurt-to-Munich link (M - F ←) has the highest throughput amongst all links in this topology, in the baseline traffic scenario, as well as in the 35% increased traffic scenario. When this link fails at baseline traffic level, its flows are diverted from Frankfurt through Hanover to Munich, causing the average throughput on the Frankfurt-to-Hanover link (H - F ←) to rise from 2.9Gbps and peak at 7.1Gbps, as can be seen in Figure 5.5c. In the 35% increase traffic scenario, the same link failure causes the traffic to be rerouted from Frankfurt through Stuttgart to Munich, raising the average throughput on the Frankfurt-to-Stuttgart link (S - F ←) from 1.7Gbps to its peak at 7.3Gbps, as shown in Figure 5.5d. When the failed link is restored, the average throughput in both scenarios return to their pre-failure levels.

30-Links Topology

In the 30-Links topology, the average throughput under baseline condition, ranges from 1.47Gbps to 4.18Gbps for the top 10 links. However a clear gab is observed between the bottom 3 of the 10 links that range between 1.47Gbps and 1.5Gbps, and the rest of the 7 links, which are between 2.7Gbps and 4.18Gbps marks, as shown in Figure 5.6a.

A traffic increase of 35% causes the average throughput to rise to a range between 1.96Gbps and 5.65Gbps, for the top 10 links, as depicted in Figure 5.6b. A similar gab between the bottom 3 and the top 7 of these links is observed in this scenario as well. However, the gab has grown slightly, from 1.21Gbps in the baseline scenario to 1.64Gbps in the increased traffic scenario.

In the second topology, the Frankfurt-to-Munich link (M - F ←) is still the link with the highest throughput, both during baseline traffic, as well as when the traffic increases by 35%. This is therefore the link that is chosen for the single-link failure analysis. When this link fails during baseline traffic, its throughput drops to zero, white that of the neighboring Frankfurt-to-Stuttgart link (S - F ←) shoots up from 2.93Gbps to 7.06Gbps, which is more than double the value before the failure. Other top 10 links are also affected and are observed to have lower throughputs during the failure. After the failed link is restored, its throughput returns to values slightly higher than what they were before the failure. The neighboring (S - F ←) link also returns to values slightly higher than the ones it had before the failure. However, not all of the top links are restored to values at or above their pre-failure levels. The

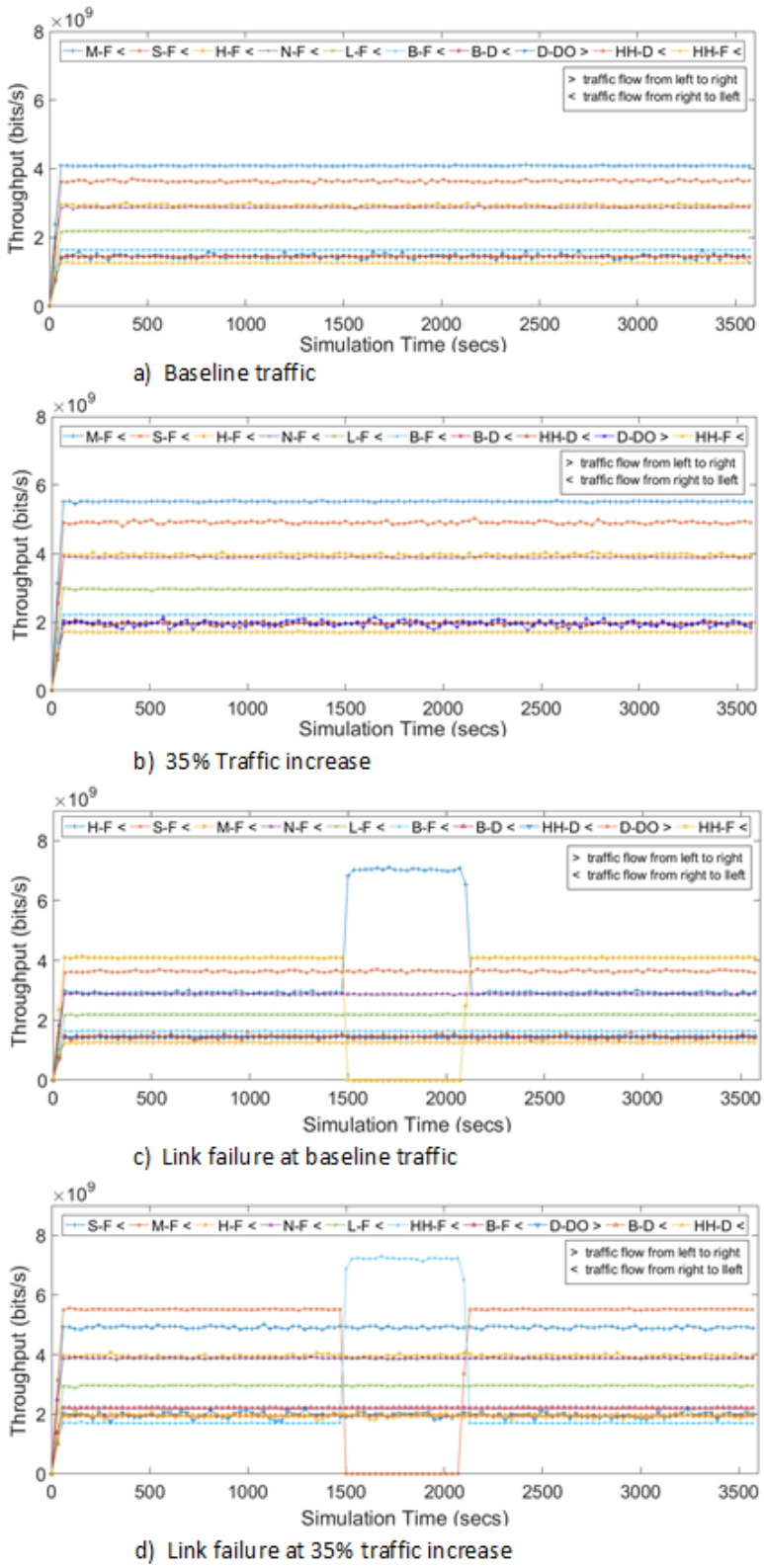


Figure 5.5: Average link throughput in Fullmesh topology (top 10)

throughput of a few of the top 10 links increase slightly, but not to their pre-failure levels, while others remain unchanged, even after the failed link is restored. These observations can be seen in Figure 5.6c.

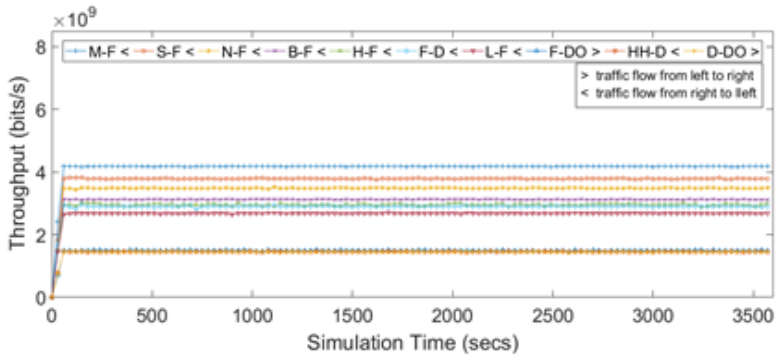
The effect of the same link failure at 35% increased traffic is a bit different and more intensified on some of the top 10 links. For example, when the Frankfurt-to-Munich link (M - F ←) fails at 35% increased traffic, most of its traffic is diverted via a different neighboring link, the Frankfurt-to-Leipzig link (L - F ←), which is also the functioning link most affected by the failure. Its throughput grows from 2.93Gbps to a peak at 7.11Gbps during the failure. This 4.18Gbps change in throughput is comparable to the 4.13Gbps change experienced by the most affected link (S - F ←) in the baseline traffic scenario.

20-Links Topology

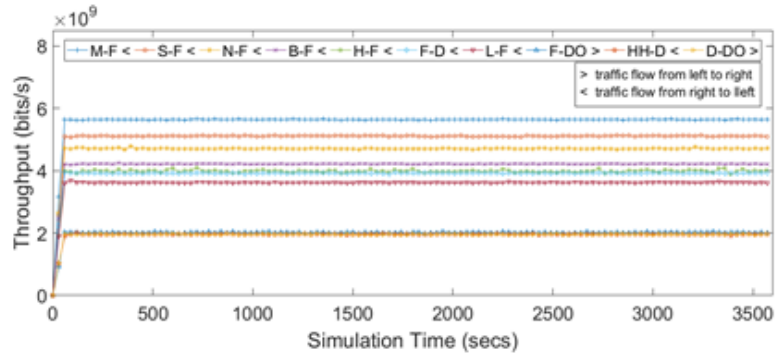
Of the 3 topologies in this study, the 20-Links topology has the least number of links, as well as the least count of the faster 10Gbps links. Under baseline traffic condition, the top 10 links record throughput values that range from 2.46Gbps to 7.64Gbps. Figure 5.7a shows that these are sub-divided into 3 sub-ranges, with the 6 of the top 10 links in the low range that goes from 2.46Gbps to 3.14Gbps, 2 in the mid-range that lies between 5.1Gbps and 5.2Gbps and 2 in the top range, which is between 7.26Gbps and 7.64Gbps.

When the traffic in the topology increases by 35%, the average throughput of the top 10 links also increases to a range between 3.34Gbps and 9.5Gbps (which is the maximum possible throughput, because of the OC-192 speed of the link). The 3 sub-ranges still exist, but are now between 3.33Gbps and 4.24Gbps for the low range, 6.86Gbps and 7.04Gbps for the two links in the mid-range and 9.5Gbps for the two links at the top-range, as is shown in Figure 5.7b. We notice a drop in average throughput on the Nuremberg-to-Munich link (N - M →), which is attributed to congestion and excessive queuing delay on the Frankfurt-to-Nuremberg link, for packets from Frankfurt destined for Munich (see Sections 5.4.3 and 5.4.4).

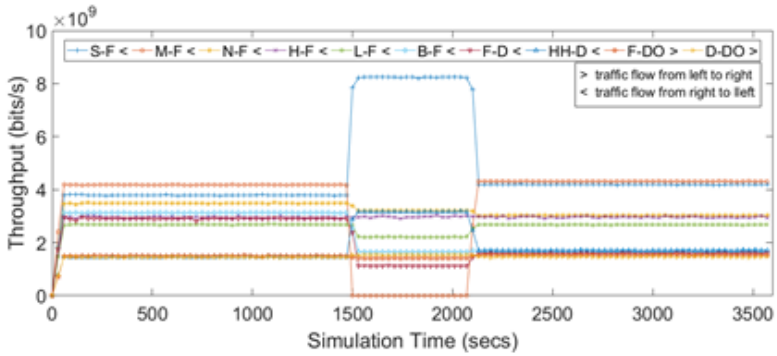
For the single link failure analyses, we use the Frankfurt-to-Nuremberg link (N - F ←), since it has the highest throughput during baseline traffic, as well as when traffic increases by 35%. The failure of this link during baseline traffic causes a spike in the average throughput of the neighboring Frankfurt-to-Leipzig link (L - F ←) that goes immediately from 2.68Gbps to its maximum at 9.5Gbps. After the failed link is restored, its average throughput increases to values a bit higher than what they were before the failure, while those of the Frankfurt-to-Leipzig link fall back to values a bit lower than what they were before the failure.



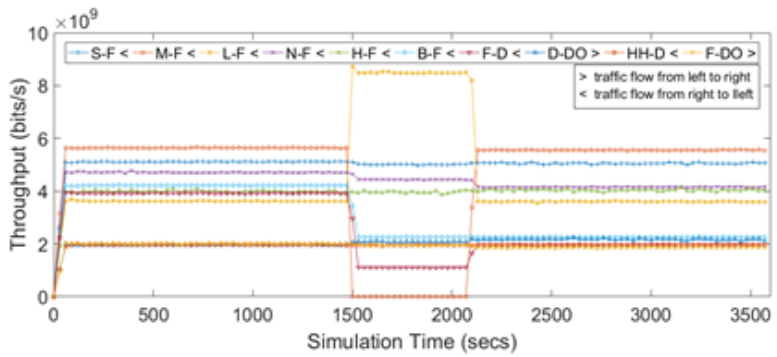
a) Baseline traffic



b) 35% Traffic increase



c) Link failure at baseline traffic



d) Link failure at 35% traffic increase

Figure 5.6: Average link throughput in 30-Links topology (top 10)

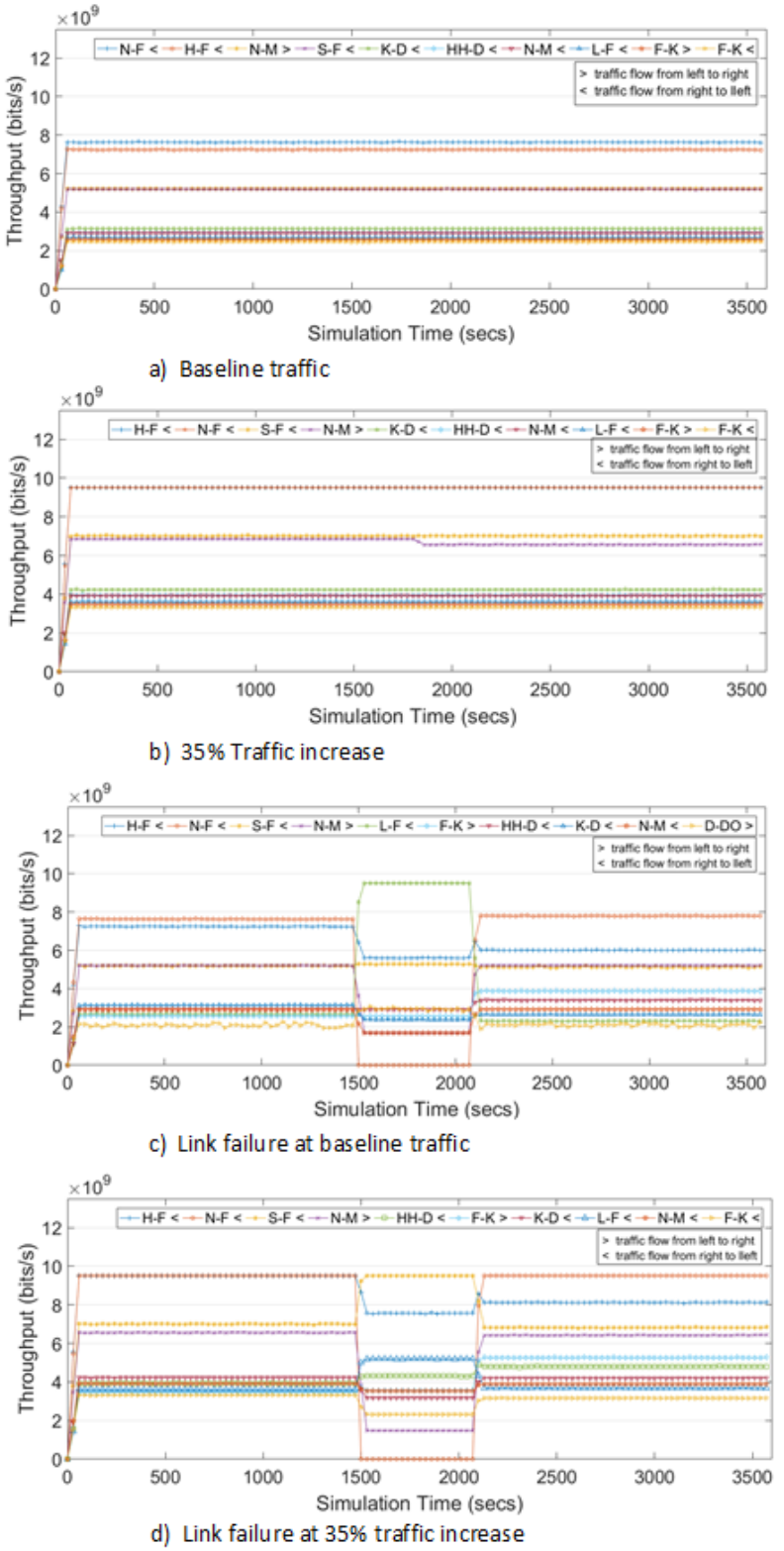


Figure 5.7: Average link throughput in 20-Links topology (top 10)

5.4.3 TCP Delay

The TCP delay provides the time lapse between sending a TCP datagram at the source node and receiving it at the destination node. This time is recorded for all TCP transactions in each topology. Figure 5.8 shows the average TCP delay per topology, for all three topologies and all 4 scenarios.

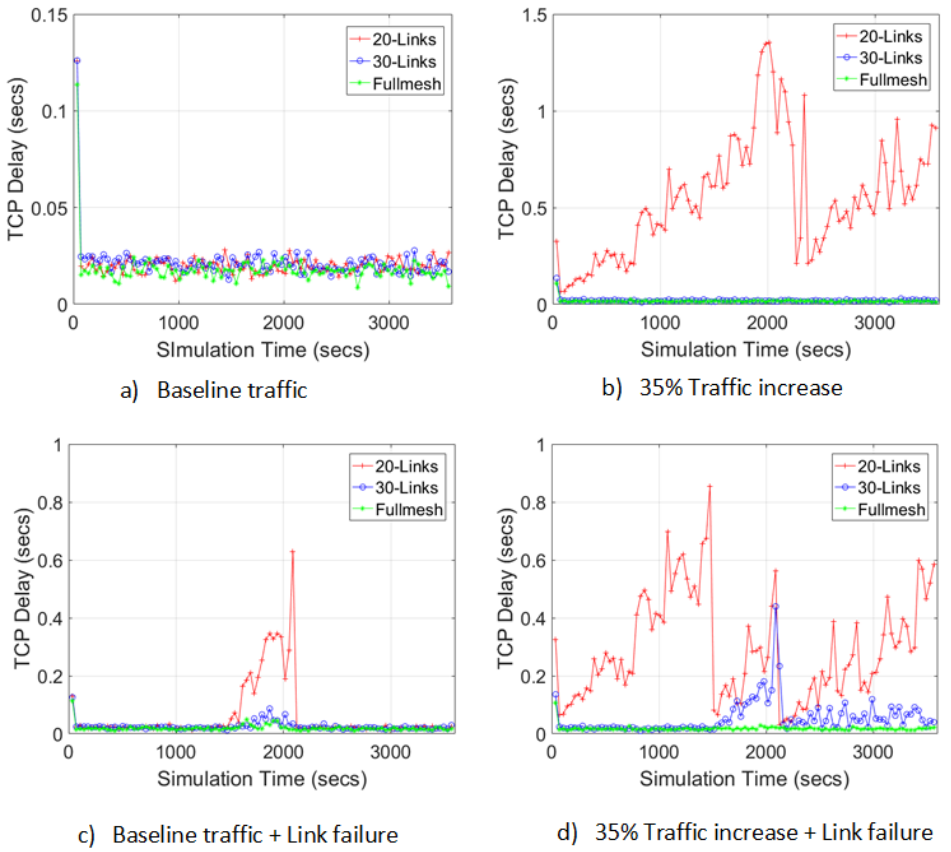


Figure 5.8: Average TCP Delay

The average TCP delay[3] is mostly below the 25ms mark for all 3 topologies during baseline traffic, see Figure 5.8a. However, in all 4 scenarios, it remains slightly better (and visibly lower) in the Fullmesh topology than in the 30-Links and 20-Links topologies.

[3]In all 4 scenarios, the peak average TCP delays at the start of the experiment are not considered, since they are recorded when some processes are still initializing. This applies to all measurements in this study.

The effects of limited bandwidth in the 20-Links topology, compared with the other two topologies, becomes obvious when the traffic increases by 35%. Its average TCP delay starts rising from the beginning up to about 1.4secs, then drops to about 213ms at the 2300s time-line before rising again from there till the end of the simulation. This behavior is attributed to queue buildup and release on two saturated links in the topology, both of which are completely maxed out in terms of throughput and utilization.

The TCP delay in the Fullmesh and 30-Links topologies remain mostly unaffected by the increased traffic, but variably affected by the link failure. At baseline traffic the failure affects the delay in both topologies equally. At increased traffic, the delay in 30-Links topology is much more affected than that in the Fullmesh topology, which shows only minor impact.

5.4.4 TCP Retransmissions

A TCP retransmission occurs when a sent TCP segment is lost (i.e. not confirmed by the destination host) and therefore needs to be resent. Continuous retransmissions are an indication of congestion in the path from the source node to the destination node.

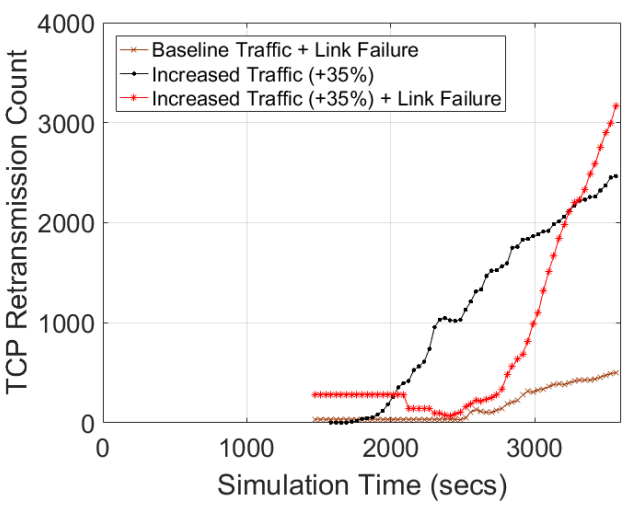


Figure 5.9: TCP Retransmissions in 20-Links topology

At baseline traffic, there are zero TCP retransmissions in all 3 topologies. When traffic in the topology is increased by 35%, no TCP retransmission is recorded in the Fullmesh and 30-Links topologies. Per contra, a significant number of TCP retransmissions are recorded in the 20-Links topology as a result of the increased load.

When the link failure occurs, the number of retransmissions recorded in the Fullmesh and the 30-Links topologies are negligibly small (11 and 32 respectively). These are also recorded as one-time-events that occur once, immediately after the failure. In contrast, the number of retransmissions in the 20-Links topology resulting from the link failure is much higher and continues even after the failed link has been restored. Figure 5.9 shows the number of TCP retransmissions resulting from congestion at 35% increased traffic and link failures in the 20-Links topology. The knock-on effect of congestion becomes noticeable in the later section of the increased traffic scenario, when the number of retransmissions goes from zero all through to a maximum of nearly 2500 at the end of the run.

5.4.5 RTP Delay

RTP delay records the time difference between the time when an RTP packet is timestamped at the source node and the time it is received at the destination node.

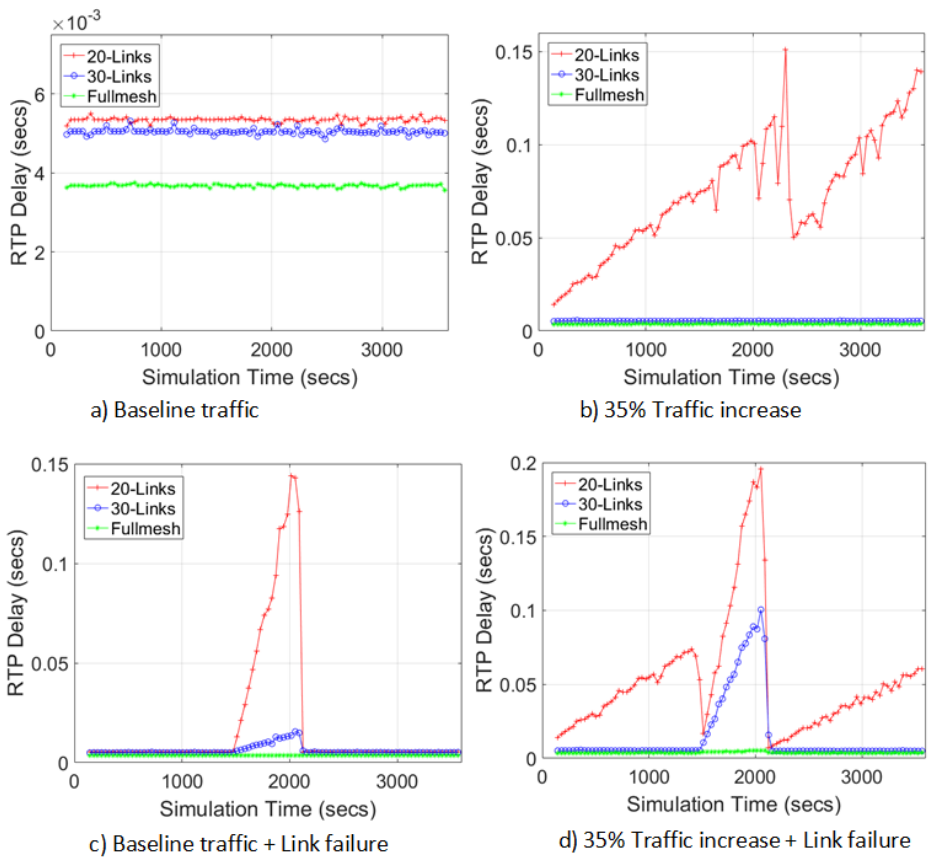


Figure 5.10: Average RTP Delays

The delay is recorded for voice packets in each topology. Figure 5.10a shows that at baseline traffic, the average RTP delays in all three topologies remain relatively low and approximately constant over time. However, the best average values of well below 4ms are recorded in the Fullmesh topology, followed by the 5.0ms and 5.3ms averages in the 30-Links and 20-Links topologies respectively.

When traffic in the topology increases by 35%, the average RTP delay in the 20-Links topology no longer remains constant. It rises right from the start, reaching a maximum of 151ms before dropping to 50ms and then rising again from there to about 139ms at the end of the simulation run. As can be seen in Figure 5.10b, the respective constant and lower average delays in the Fullmesh and the 30-Links topologies are maintained, despite the increased traffic load.

Link failure has adverse effects on the average RTP delay in the 20-Links topology, as can be seen in Figures 5.10c and 5.10d respectively. The failure at baseline traffic causes a sharp increase in average RTP delay from a few milliseconds to a maximum of approximately 144ms. This falls back to normal values immediately after the link is restored. In the 30-Links topology, only a slight increase in delay occurs, as a result of the link failure. This also falls back to normal after the failed link is restored. Only negligible changes are observed in the Fullmesh topology. Neither the link failures nor the increased traffic in the topology shows any significant effect on the delay values.

5.5 Application Performance Analyses

In this section, we analyze and compare the performance of some standard applications, with regards to traffic conditions in each of the 3 selected topologies.

5.5.1 FTP Download Response Time

The FTP Download response time is the timespan between sending an FTP request to a server and receiving the complete response packet from it. The average response time in each topology is shown in Figure 5.11. At baseline traffic, Figure 5.11a shows only slight difference between the average download times in each of the 3 topologies. However, when link failure occurs, Figure 5.11c shows a huge increase in the average download time in the 20-Links topology, compared to only slight changes in the 30-Links and Fullmesh topologies.

When traffic in the topology increases by 35%, Figure 5.11b shows that the average download times in the 30-Links and Fullmesh topologies remain continuously low, while that in the 20-Links topology increases from the start to a maximum of 49.98 seconds, after 2232 seconds of simulation. It then drops to 5.23 seconds shortly thereafter, before slowly rising again with time.

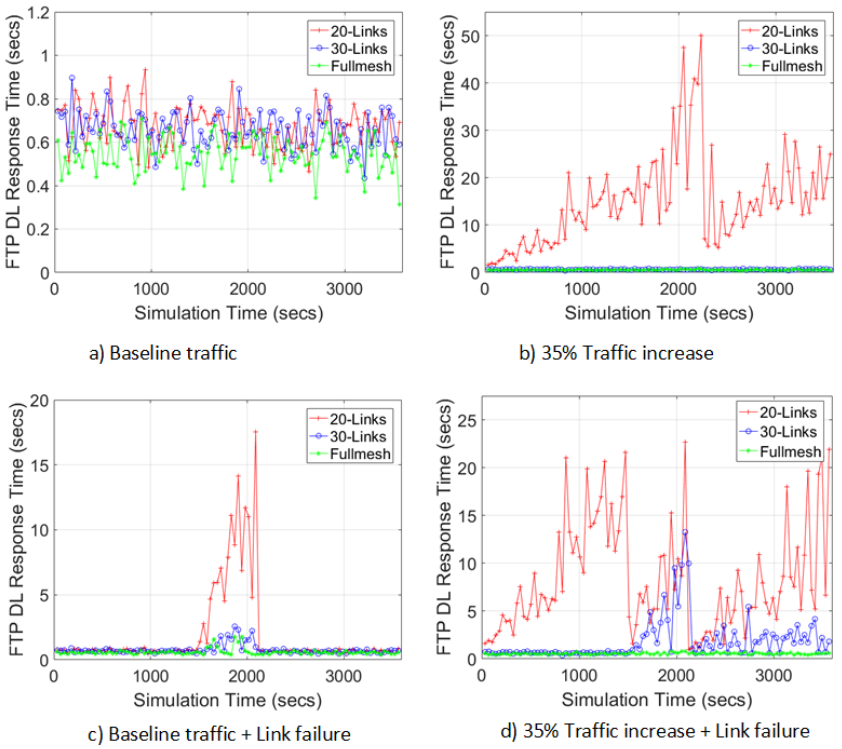


Figure 5.11: Average FTP Download Response time

Link failure at this increased traffic load, only minimally affects the download times in the Fullmesh topology, as can be seen in Figure 5.11d. The average FTP download time in the 30-Links topology clearly increases as a result of the failure, while that of the 20-Links topology first drops and then starts increasing again during the failure period. It drops again after the link is restored and then rises again immediately thereafter.

5.5.2 HTTP Received Traffic

In all three topologies, we observe that the amount of HTTP traffic that is received in bytes/s, closely corresponds to the amount that is sent[4]. At baseline traffic all 3 topologies send and receive approximately the same amount of HTTP traffic, as can be seen in Figure 5.12a for received traffic only. However, when the total traffic in the topology increases by 35%, a stark difference between the topologies is observed. Under this condition, much less HTTP traffic is sent and received in the 20-Links topology than in the Fullmesh and 30-Links topologies (see Figure 5.12b).

[4]Since the amount of HTTP traffic receieved closely corresponds to the amount sent, only the amount receieved is plotted

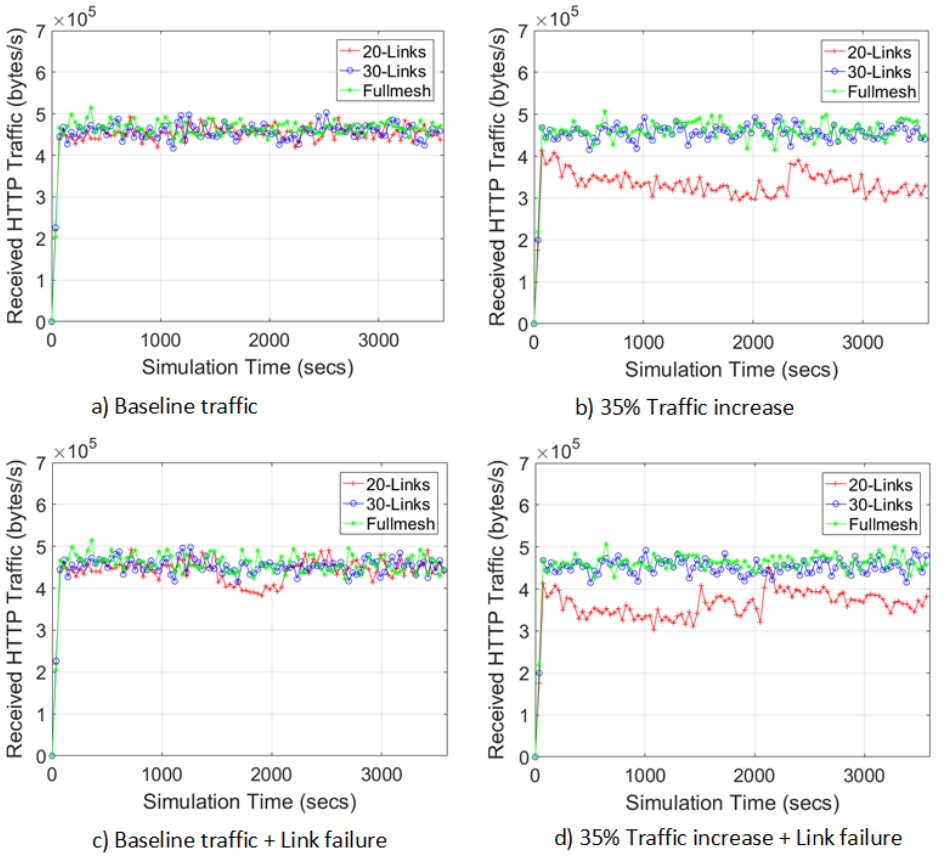


Figure 5.12: Average HTTP Traffic Received

The effect of a single link failure on the quantity of HTTP traffic sent and received in each topology, at baseline traffic, as well as at increased traffic, can be seen in Figure 5.12c and Figure 5.12d respectively. While both quantities are virtually unaffected by the link failures in the Fullmesh and 30-Links topologies respectively, in the 20-Links topology and at baseline traffic, we notice a significant drop in the quantity received during the link failure. When link failure occurs at increased traffic load, the quantity of HTTP traffic received is shown to slightly increase and then slowly drop again with time.

5.5.3 HTTP Object Response Time

The HTTP object response time is the time taken by a browser (or an HTTP client) to retrieve an HTTP embedded object from an HTTP server. All HTTP transactions are taken into consideration when calculating the average object response time in a given topology.

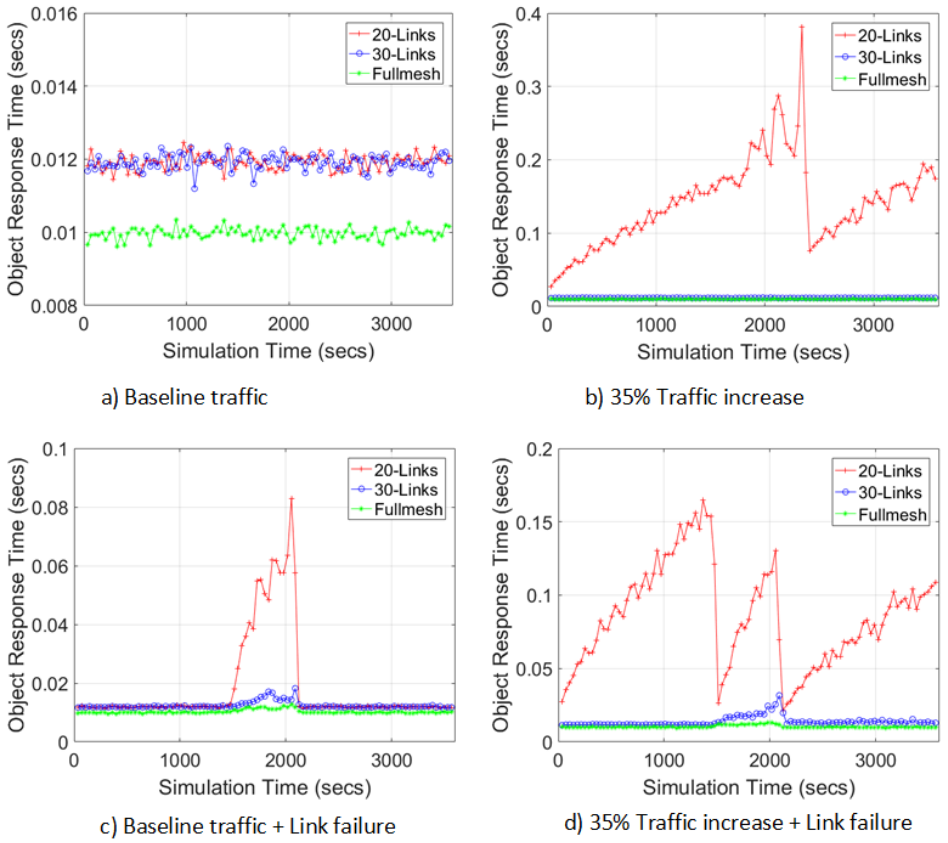


Figure 5.13: Average HTTP Object Response time

At baseline traffic, Figure 5.13a shows a similar average HTTP object response time of approximately 12 milliseconds in the 20-Links and 30-Links topologies. The Fullmesh topology records an average of 10 millisecond, which is, 2 milliseconds better than in the 20-Links and 30-Links topologies. Figure 5.13c however shows that, when the link failure occurs, the average response time in the 20-Links topology increases the most, to nearly 83 milliseconds. In the 30-Links and Fullmesh topologies, it increases only to 17 and 12 milliseconds respectively.

At increased traffic load, Figure 5.13b and Figure 5.13d, show that only the average HTTP object response time in the 20-Links topology is adversely affected by the increased load and the link failure respectively. The average response time in the 30-Links and Fullmesh topologies remain low under the same conditions or are only minimally affected by the link failure.

5.5.4 Voice Packet Jitter

Voice packet jitter is a measure of the variation in the delay of received voice packets. If two consecutive packets leave the source node with time stamps t_1 & t_2 and are played back at the destination node at time t_3 & t_4, then:

$$jitter = (t_4 - t_3) - (t_2 - t_1)$$

A negative jitter indicates that the time difference between the packets at the destination node was less than that at the source node. The recommended tolerance for one-way peak-to-peak voice packet jitter is 30ms or less [199].

At baseline traffic, the jitters in all 3 topologies are in the low nanosecond-range and therefore negligible (Figure 5.14a).

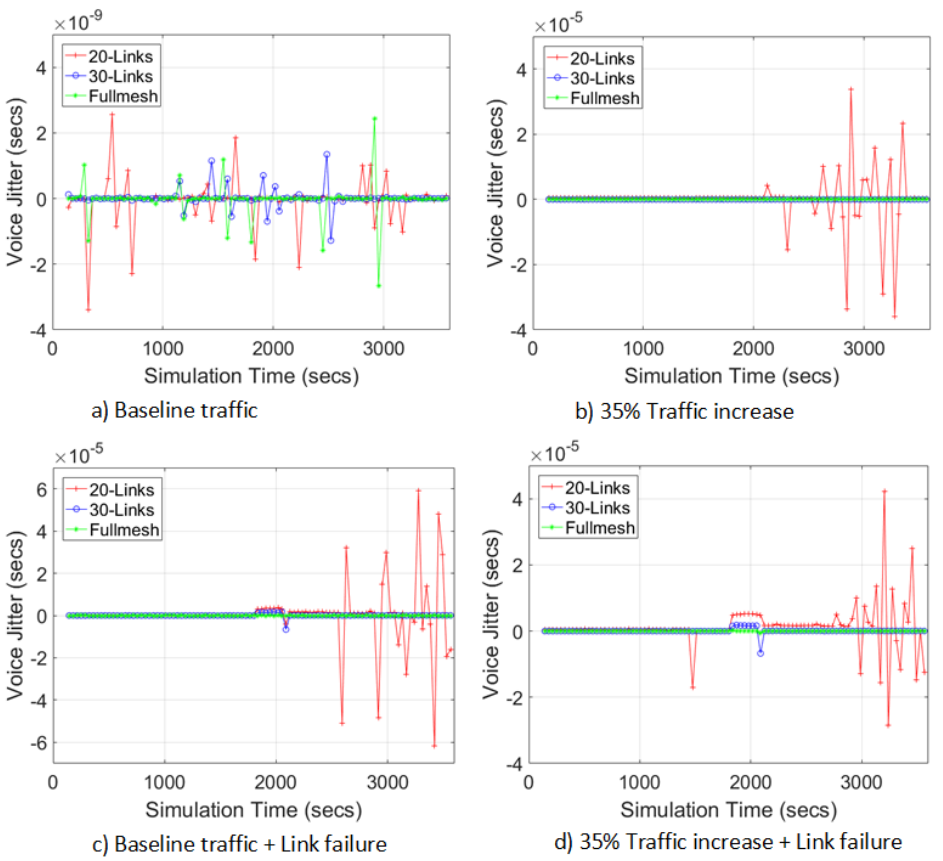


Figure 5.14: Voice Jitter

With increased traffic, the jitter in the Fullmesh and the 30-Links topologies remain virtually unchanged in the nanosecond range, from start to end. An increase in jitter is however observed in the 20-Links topology later into the simulation. Although its measured jitter has increased to an order of magnitude in the lower-10s of microseconds, the values are still negligible when compared with the acceptable tolerance of 30 milliseconds. The risen jitter is nevertheless still indicative of increased delay (or congestion) resulting from the increased load in the topology.

All 3 topologies show slight but negligible increase in the jitter during link failure. However only the 20-Links topology seem to suffer from an after-effect, long after the failed link has been restored, as can be seen in Figure 5.14c and 5.14d.

When the link failure occurs at increased traffic load, the jitters in the Fullmesh and the 30-Links topologies still remain in nanoseconds-range, while those in the 20-Links topology also remain in the lower-10s of microseconds-range. The 35% increase in traffic, combined with the failure of the top-most link in each of these topologies, do not seem to affect their jitters in any negative way. This is supported by the graph in Figure 5.14d, which shows that the values recorded for their jitters always remain far below the acceptable tolerance value of 30 milliseconds.

5.5.5 Video Packet End-to-End Delay

The Video Packet end-to-end delay measures the time taken to send a video application packet from a source node to a destination node. Although average delays might differ, depending on the physical properties of the topology, it is important that they remain constant in value, for good viewing experience. Real-time interactive video tolerates packet end-to-end delays of up to 200ms and jitters of up to 50ms [199] for High Definition (HD) flows. At baseline traffic, the video packet end-to-end delay in all three topologies have constant averages of 4.5ms, 5.4ms and 5.9ms for the Fullmesh, 30-Links and 20-links topologies respectively. These are quite good values, as can be seen from Figure 5.15a.

When traffic in the topology is increased, the video packet end-to-end delay in the Fullmesh and 30-Links topologies still remain low while that of the 20-Links topology increases right from the start. This indicates the presence of congestion in the 20-Links topology (Figure 5.15b).

Link failure at baseline traffic causes a noticeable increase in video packet end-to-end delay in all 3 topologies (Figure 5.15c). However, only the delay in the 20-Links topology rises to 58 ms, which, though is higher than those in the other 2 topologies (both below 20ms), is still well within the acceptable delay tolerance for HD video flows.

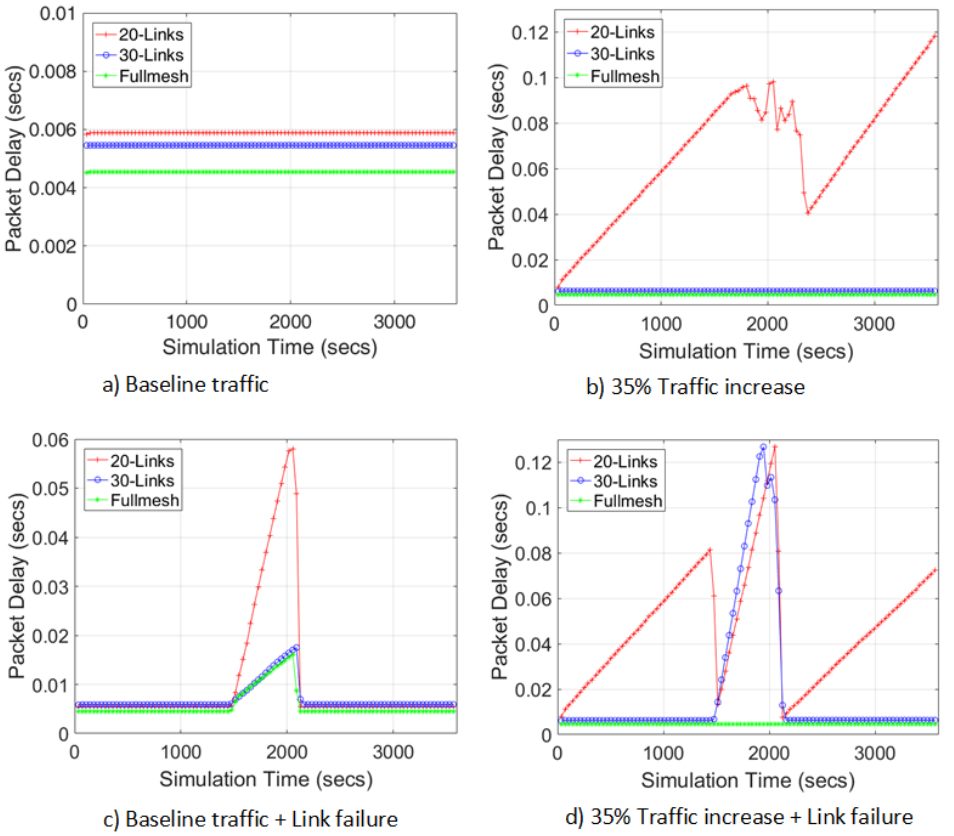


Figure 5.15: Video Packet End-to-End Delay

A combination of increased traffic and link failure in the 20-Links topology causes the already high video packet end-to-end delay value to first drop at the time of the failure and then rise rapidly again until restoration. Immediately after the restoration, it first drops and then starts increasing. This reveals a higher level of congestion in the topology, which affects the active path, as well as the backup paths that the video packets take when the primary link fails. In the 30-Links topology, the delay increases the moment the link fails, but falls back to normal low values immediately after the link is restored. The delay in the Fullmesh topology is minimally affected by the failure, as can be seen in Figure 5.15d.

5.6 Summary

To study how different backbone topologies affect the general performance in a network, we design 3 different backbone topologies using a national reference backbone model for Germany. We keep the number and location of the nodes constant but vary

the number and distribution of their interconnecting links, to obtain 3 topologies that differ in structural characteristics and total bandwidth capacities. We name them Fullmesh (with 66 links), 30-Links and 20-Links topology respectively.

We carry out simulation studies on all 3 topologies, placing each of them under the same network/traffic conditions. These conditions are represented in the simulation environment by the following defined scenarios i) baseline traffic ii) baseline traffic + single link failure iii) 35% increased traffic iv) 35% increased traffic + single link failure. We use selected network and application metrics to analyze and compare the performances in each of the 3 modeled topologies and for each of the defined scenario.

All 3 topologies show comparative performance in only a few cases during baseline traffic. However, for many of the selected metrics and traffic conditions, the performances in the 30-Links topology comes much closer to those in the Fullmesh topology. Nevertheless, clear differences become obvious when traffic in the topology increases by 35% and/or when a link failure occurs. The 20-Links topology is observed to offer the least performance of all 3 topologies under these conditions, while the best performance in nearly all scenarios are recorded in the Fullmesh topology.

6

Flow Optimization using Mixed-Integer Programming (MIP)

In this chapter, we exploit the use of Mixed-Integer Programming to improve the performance in the 20-Links topology, which is observed in the previous chapter, to offer the least performance amongst the 3 that are compared. Some sections of its network are observed to be highly congested while others literally have little-to-no traffic on them. We think altering the distribution of load in this topology and minimizing the maximum utilization on its links should lead to performance improvements. We therefore investigate via simulations if this postulation holds true under the stated conditions. For this, we employ flow optimization as a means of efficiently distributing the traffic load in the network, without changing its architecture or physical topology. We plan, run and analyze two sets of experiments; one set using OSPF interface costs obtained via automatic (default) metric calculations, the other set using OSPF interface costs obtained via flow-optimized Mixed-Integer Programming.

There are a good number of works on optimizing intra-domain routing weights to achieve desired traffic engineering goals [66–69]. In this thesis and particularly in this chapter, we aim at the same goals, but differ in our approach and assumptions. While most of the former works depend on flow-splitting and mostly linear programming approaches, we follow a more realistic approach that takes *unsplittable* "elephant" flows into consideration [27] and use MIP instead of LP, to enable us include both linear and non-linear constraints in our problem formulation.

6.1 MIP Problem Formulation

To determine the metrics needed for optimized flow in the network, we formulate an MIP problem with appropriate constraints.

We consider the directed graph G = (V, E), where V represents a set of vertices (or PoPs) and E a set of edges (or links). The capacity is given by $c : E \to \mathbb{R}$ and the demand matrix by $D : V \times V \to \mathbb{R}$, where $D(i, j)$ denotes the demand flowing from PoP i to PoP j via the path P_{ij}. The task is to maximize the bandwidth used in

the network subject to the following conditions: i) the OSPF routing and ii) edge capacities are not violated. We define the following variables:

$w_e \in \mathbb{N}$	OSPF weight of edge $e \in E$
$x_{P_{ij}^k} \in \{0,1\}$	decides whether path $P_{ij}^k \in \mathcal{P}_{ij}$ can be used for transmission
$f_{P_{ij}^k} \in \mathbb{R}^+$	amount of flow (bandwidth) sent from i to j in the *k-th* path
$cost_{P_{ij}^k} \in \mathbb{R}^+$	OSPF cost of the *k-th* path from i to j
M	maximum OSPF weight
$c(e)$	capacity of edge (or link) e
U_{max}	maximum utilization on link

Our objective is to minimize the maximum utilization on the links in order to avoid congestion.

$$minimize\ U_{max} \tag{6.1}$$

subject to the following constraints:

$$cost_{P_{ij}^k} = \sum_{e \in P_{ij}^k} w_e \qquad \forall (i,j) \in V \times V : \forall P_{ij}^k \in \mathcal{P}_{ij} \tag{6.2}$$

$$min\_cost_{ij} \leq cost_{P_{ij}^k} \qquad \forall (i,j) \in V \times V : \forall P_{ij}^k \in \mathcal{P}_{ij} \tag{6.3}$$

$$cost_{P_{ij}^k} - min\_cost_{ij} \leq (1 - x_{P_{ij}^k}) \cdot M \qquad \forall (i,j) \in V \times V : \forall P_{ij}^k \in \mathcal{P}_{ij} \tag{6.4}$$

$$\sum_{P_{ij}^k \in \mathcal{P}_{ij}} f_{P_{ij}^k} = D(i,j) \qquad \forall (i,j) \in V \times V \tag{6.5}$$

$$f_{P_{ij}^k} \leq D(i,j) \cdot x_{P_{ij}^k} \qquad \forall (i,j) \in V \times V : \forall P_{ij}^k \in \mathcal{P}_{ij} \tag{6.6}$$

$$U_{max} \geq \frac{\sum_{(i,j) \in V \times V} \sum_{P_{ij}^k \in \mathcal{P}_{ij} : e \in P_{ij}^k} f_{P_{ij}^k}}{c(e)} \qquad \forall e \in E \tag{6.7}$$

$$c(e) \geq \sum_{(i,j) \in V \times V} \sum_{P_{ij}^k \in \mathcal{P}_{ij} : e \in P_{ij}^k} f_{P_{ij}^k} \qquad \forall e \in E \tag{6.8}$$

6.2 Solving the MIP Problem

MIP problems are generally much harder to solve than e.g. LP problems. The time taken to solve an optimization problem depends on many factors, such as the size of the topology, the numbers of involved parameters, variables and constraints, as well as the speed of the programming software (solver). Using modern computing systems that have very fast processing units and large memories, the time needed to solve such problems could be kept relatively short. There are two major categories of solvers available to users, i) free/open-source solvers and ii) commercial solvers. We started off using the GNU Linear Programming Kit (GLPK) [138], the most popular free solver, but quickly noticed it was too slow and could not solve our problem in reasonable time. We thus checked on the three most renounced commercial solvers for MIP, i.e. IBM CPLEX [41], FICO XPRESS [55] and Gurobi [155]. We chose the *Gurobi Optimizer* [154] because of its superior speed compared with all the other solvers [141, 142], its reputation within the industry and the fact that they offer a free and full-featured academic license. With it, we could obtain solutions within minutes, compared to hours when using the free/open-source solvers.

6.3 Simulation Study

To perform the simulation studies in this chapter, we use the same 20-Links topology that is used in the previous chapter and increase the traffic in the topology by 35%, as is done in the increased-traffic scenarios of the previous chapter. We next design and run two sets of experiments using two different routing schemes. One based on automatically calculated cost metric (the default method), the other using MIP to determine optimized cost metric values for each router interface attached to an active link.

The first set of experiments are based on default interface costs. Their results are denoted by *AUTO* because of the automatic calculation of the metric using the default formula:

$$\text{Interface cost} = \frac{\text{Reference Bandwidth}}{\text{Interface Bandwidth}}$$

We use 100Gbps as our reference bandwidth instead of the default 100Mbps. This is necessary to overcome the limitations of the default 100Mbps when dealing with links of much higher bandwidths. Leaving the reference bandwidth at 100Mbps would yield the same metric value for all links with speed greater than or equal to 100Mbps, which is not appropriate for our purposes.

The second set of experiments use interface costs that are determined via the MIP optimization process. Its results are denoted with *OPT*. It should be noted here

that the process to determine the OPT cost metrics is done externally, in a Gurobi standalone environment. The results are then collected and manually entered into the OPNET simulation environment. In a real production environment, this task could be automated, using an appropriate Network Management System (NMS) that, for example, ensures routing loops and disruptions are avoided [70].

In both sets of experiments, neither the topology nor architecture is changed. The only change involved is that of adjusting the OSPF routing metrics to influence the paths that flows take.

We simulate two scenarios each for the AUTO routing topology and the OPT routing topology, respectively. That is, a normal scenario (increased traffic without link failure) and a failure scenario (increased traffic with single link failure). For the single link failure, the link with the highest throughput and utilization during normal operation (baseline traffic) is selected and failed for 10 minutes in the middle of the simulation run, i.e. in the period between 1500 and 2100 seconds.

6.4 Results and Analyses

Since no topological changes are involved in the scenarios studied in this chapter, all results and analyses will be based on application and protocol performances only.

6.4.1 Throughput and Utilization

Although throughput and utilization are important link performance and comparison metrics, they take on different significance when analyzing/comparing links of unequal capacities. In the case of backbone topologies, throughput takes on a more important significance because of the need to push through the largest possible volumes of data across a network at the highest possible speed. Utilization plays the role of an indicator on individual links. It can be used to signal (indicate) when a link becomes saturated, warranting an upgrade or a change of routing policy to offload traffic from it.

We compare the throughput and the utilization on the individual links in the topology when routing is based on AUTO interface costs and OPT interface costs respectively. Both directions of flow on a link are considered/analyzed separately (as in a directed graph). Table 6.1 shows the throughput and utilization in both directions of a link.

The results for the AUTO routing topology are shown in Table 6.1a. We observe that links with higher bandwidths are also those with higher throughputs. This is expected, since the routing metric favors paths with high-bandwidth links over those with lower-bandwidth links. The results for the OPT routing topology are shown in Table 6.1b.

Link Name	BW (Mbps)	Util. Fwd (%)	Tput Fwd (Mbps)	Util. Rtn (%)	Tput Rtn (Mbps)	Tput (Fwd + Rtn) (Mbps)
N <-> F	9,510.91	48.73	4634.564	100.09	9519.702	14,154.266
H <-> F	9,510.91	41.13	3911.639	100.11	9521.491	13,433.13
N <-> M	9,510.91	68.34	6499.547	40.75	3875.685	10,375.232
S <-> F	9,510.91	16.6	1578.339	78.71	7486.138	9,064.477
F <-> K	9,510.91	57.36	5455.121	35.9	3414.27	8,869.391
K <-> D	9,510.91	40.08	3812.347	47.53	4520.767	8,333.114
B <-> H	9,510.91	27.47	2612.884	57.59	5477.674	8,090.558
HH <-> H	9,510.91	31.44	2989.983	51.94	4939.549	7,929.532
HH <-> D	9,510.91	50.87	4838.301	11.86	1127.729	5,966.03
L <-> F	9,510.91	15.5	1473.91	46.74	4445.252	5,919.162
D <-> DO	9,510.91	42.97	4086.881	5.09	483.802	4,570.683
ULM <-> S	2,377.73	16.41	390.247	53.99	1283.797	1,674.044
M <-> ULM	2,377.73	6.39	151.993	3.06	72.814	224.807
N <-> S	2,377.73	0.13	3.074	0.14	3.239	6.313
L <-> N	2,377.73	0.11	2.538	0.09	2.199	4.737
B <-> L	2,377.73	0.03	0.689	0.04	0.947	1.636
H <-> L	2,377.73	0	0.023	0	0.051	0.074
HH <-> B	2,377.73	0	0.002	0	0.002	0.004
H <-> DO	2,377.73	0	0.001	0	0.001	0.002
K <-> DO	2,377.73	0	0.001	0	0.001	0.002
Total	**126,019.58**	**33.68**	**42,442.084**	**44.58**	**56,175.11**	**98,617.194**

a) With AUTO interface costs

Link Name	BW (Mbps)	Util. Fwd (%)	Tput Fwd (Mbps)	Util. Rtn (%)	Tput Rtn (Mbps)	Total Tput (Mbps)
N <-> F	9,510.91	36.79	3498.761	98.96	9412.424	12911.185
H <-> F	9,510.91	30.86	2934.596	74.1	7047.659	9982.255
S <-> F	9,510.91	13.35	1270.02	73.93	7031.427	8301.447
HH <-> H	9,510.91	8.96	851.735	66.94	6366.993	7218.728
ULM <-> S	2,377.73	17.2	408.968	53.19	1264.731	1673.699
B <-> H	9,510.91	14.41	1370.993	51.11	4860.593	6231.586
N <-> M	9,510.91	74.33	7069.73	40.56	3857.435	10927.165
L <-> F	9,510.91	6.53	621.381	32.72	3111.543	3732.924
L <-> N	2,377.73	87.75	2086.559	27.82	661.426	2747.985
K <-> D	9,510.91	36.92	3511.865	26.02	2474.333	5986.198
H <-> L	2,377.73	84.4	2006.719	25.85	614.593	2621.312
F <-> K	9,510.91	75.87	7215.921	16.93	1610.597	8826.518
N <-> S	2,377.73	20.05	476.685	15.51	368.833	845.518
B <-> L	2,377.73	35.91	853.899	14.3	339.972	1193.871
HH <-> B	2,377.73	16.31	387.857	11.65	276.938	664.795
HH <-> D	9,510.91	86.22	8200.049	10.85	1032.058	9232.107
K <-> DO	2,377.73	86.18	2049.235	9.75	231.869	2281.104
H <-> DO	2,377.73	2.09	49.807	6.63	157.676	207.483
M <-> ULM	2,377.73	7.2	171.189	2.28	54.274	225.463
D <-> DO	9,510.91	21.47	2042.151	1.56	148.361	2190.512
Total	**126,019.58**	**37.36**	**47078.12**	**40.41**	**50923.735**	**98001.855**

b) With OPT interface costs

Table 6.1: Link Utilization and Throughput at increased traffic

Link Name	BW (Mbps)	Util. Fwd (%)	Tput Fwd (Mbps)	Util. Rtn (%)	Tput Rtn (Mbps)	Tput Fwd + Rtn (Mbps)
N <-> S	2,377.73	100.22	2382.977	100.21	2382.816	4765.793
N <-> F	9,510.91	48.74	4635.41	100.16	9525.793	14161.203
H <-> F	9,510.91	41.14	3912.455	100.13	9523.084	13435.539
L <-> F	9,510.91	33.29	3166.515	100.06	9516.558	12683.073
S <-> F	9,510.91	35.84	3408.739	100.06	9516.712	12925.451
L <-> N	2,377.73	100.15	2381.369	94.3	2242.109	4623.478
B <-> H	9,510.91	27.48	2613.255	58.88	5599.878	8213.133
ULM <-> S	2,377.73	16.41	390.194	53.99	1283.64	1673.834
HH <-> H	9,510.91	34.4	3271.919	51.92	4937.966	8209.885
K <-> D	9,510.91	43.8	4166.107	47.52	4519.299	8685.406
N <-> M	9,510.91	95.78	9109.698	40.75	3875.864	12985.562
F <-> K	9,510.91	61.39	5838.709	35.88	3412.772	9251.481
HH <-> D	9,510.91	50.84	4835.153	11.83	1125.181	5960.334
D <-> DO	9,510.91	44.17	4200.772	5.08	483.558	4684.33
M <-> ULM	2,377.73	6.39	151.995	3.06	72.764	224.759
B <-> L	2,377.73	0.02	0.429	0.02	0.409	0.838
H <-> B	2,377.73	0	0.002	0	0.002	0.004
H <-> L	2,377.73	0	0.004	0	0.001	0.005
H <-> D	2,377.73	0	0.001	0	0.001	0.002
K <-> DO	2,377.73	0	0.001	0	0.001	0.002
Total	**126,019.58**	**43.22**	**54465.704**	**53.97**	**68018.408**	**122484.112**

a) With AUTO interface costs

Link Name	BW (Mbps)	Util. Fwd (%)	Tput Fwd (Mbps)	Util. Rtn (%)	Tput Rtn (Mbps)	Total Tput (Mbps)
S <-> F	9,510.91	40.74	3,874.42	100.05	9,515.78	13,390.20
F <-> N	9,510.91	34.96	3,324.94	99.39	9,453.09	12,778.03
H <-> F	9,510.91	28.00	2,663.45	86.20	8,198.08	10,861.53
HH <-> H	9,510.91	26.35	2,506.13	52.25	4,969.41	7,475.54
N <-> M	9,510.91	97.32	9,256.06	40.73	3,873.38	13,129.44
L <-> F	9,510.91	17.36	1,651.38	32.72	3,111.55	4,762.93
B <-> H	9,510.91	3.18	302.84	32.21	3,063.02	3,365.86
N <-> S	2,377.73	100.14	2,381.15	100.18	2,381.96	4,763.11
HH <-> B	2,377.73	18.65	443.51	93.87	2,231.96	2,675.46
K <-> D	9,510.91	23.19	2,205.46	20.14	1,915.36	4,120.81
L <-> N	2,377.73	86.94	2,067.29	56.66	1,347.14	3,414.44
H <-> L	2,377.73	84.40	2,006.69	54.63	1,298.96	3,305.65
ULM <-> S	2,377.73	16.47	391.52	53.98	1,283.42	1,674.94
HH <-> D	9,510.91	78.86	7,500.32	12.09	1,150.26	8,650.58
F <-> K	9,510.91	61.34	5,834.17	8.39	797.92	6,632.10
H <-> DO	2,377.73	0.00	0.05	14.48	344.26	344.31
B <-> L	2,377.73	78.41	1,864.42	14.35	341.14	2,205.56
D <-> DO	9,510.91	22.07	2,099.42	1.56	148.37	2,247.79
M <-> ULM	2,377.73	6.46	153.58	3.06	72.78	226.36
K <-> DO	2,377.73	89.38	2,125.30	1.83	43.60	2,168.90
Total	**126,019.58**	**41.78**	**52652.077**	**44.07**	**55541.436**	**108193.513**

b) With OPT interface costs

Table 6.2: Link Utilization and Throughput at increased traffic and single link failure

The inclusion of additional constraints apart from only bandwidth when determining the interface cost, means that paths having high-bandwidth links are not necessarily preferred over those having lower-bandwidth links.

In addition, the optimization process that also takes into consideration factors such as limitation of the maximum utilization on links and sharing of load across multiple paths, can cause some lower-bandwidth links to experience higher throughputs than some high-bandwidth links.

In terms of utilization, we observe 2 over-utilized[1] links in the AUTO routing topology, as shown in Table 6.1a. One from F to N and the other from F to H. In contrast, there are no over-utilized links in the OPT routing topology, as can be seen from Table 6.1b, because of the same reasons mentioned in the previous paragraph.

In the AUTO routing topology we observe a poor distribution of traffic, with many links having little or no traffic on them, while others are close to saturation or even saturated. In the OPT scenario, we observe a better distribution of traffic across all links.

With link failure we observe over-utilization on 6 links in the AUTO routing topology, as can seen in Table 6.2a and only on 2 links in the OPT routing topology, as shown in Table 6.2b.

6.4.2 Packet Hop Count

Again, we consider only packets attributed to remote connections to/from PoPs other than the local PoP, i.e. packets that are transmitted via backbone links.

Hop Count	3	4	5	6	7	8	9
AUTO-cost Scenario	12,223,537	10,772,726	9,054,909	3,849,866	506,038	101,983	81,408
AUTO-cost Scenario (Link Failure)	12,122,400	11,355,625	8,959,636	4,098,311	935,523	180,923	96,430
OPT-cost Scenario	16,866,970	15,390,016	6,016,176	4,563,686	868,525	0	0
OPT-cost Scenario (link Failure)	18,472,267	12,684,168	7,133,082	2,901,424	502,175	0	0

Table 6.3: Packets Hop count at 35% increased traffic (20-Links Topology)

Table 6.3 shows the number of hops that these packets traverse before arriving at their final destination and the total count of packets in each case. For each topology, both the normal, as well as the failure scenarios are shown. The same information is presented as bar charts in Figure 6.1.

[1] For our experiments, we consider utilizations up to 99.9% as normal. Any value above this limit is considered over-utilization

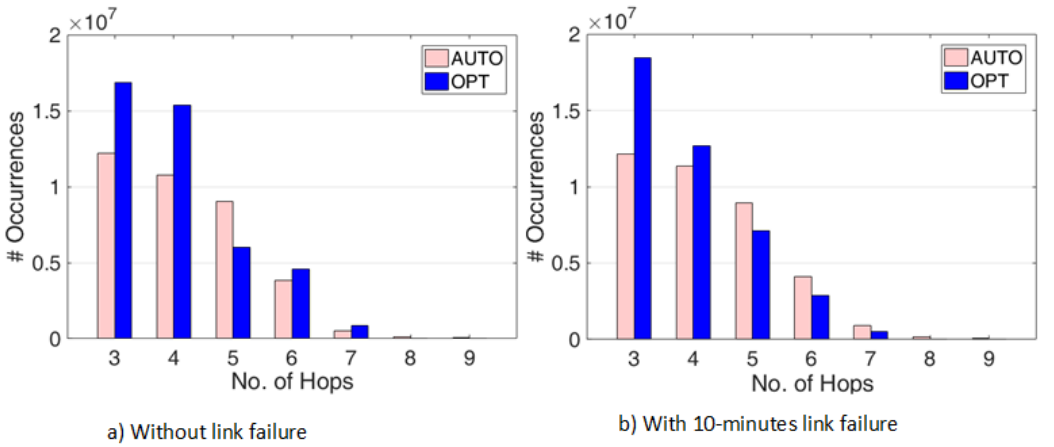


Figure 6.1: Packet hop count at 35% increased traffic (Bar chart)

First, we notice that, while some packets in the AUTO routing topology travel right up to 9 hops and beyond, there are none in the OPT routing topology that travel beyond 7 hops, even when link failure occurs.

Although, in all scenarios, the packet count decreases as the number of hops increases, for hops number 3 and 4, during normal operations, we observe a large difference in packets-count between the AUTO and the OPT topologies, as shown in Figure 6.1a. In the presence of failure, this difference increases even more for hop number 3, while it reduces for hop number 4, as shown in Figure 6.1b.

As was observed in Section 6.4.1 and shown in Table 6.1 and Table 6.2, there are many links in the AUTO topology that have little or no traffic on them. This is in contrast to the OPT topology, where there is a better distribution of traffic across its links, leaving virtually no link unused. For the unused links in the AUTO topology, the same links are shown to be in use in the OPT topology to transfer even more data between the PoPs on both ends of the links. This accounts for the observed difference in the number of packets with low hop counts, especially for hops 3 and 4 as can be seen in Figure 6.1.

6.4.3 TCP Performance

We next analyze the performance of TCP, based on three criteria; the TCP delay, the TCP segment delay and the number of TCP retransmissions. For each of these criteria, we analyze their performance under normal condition, as well as under link failure condition.

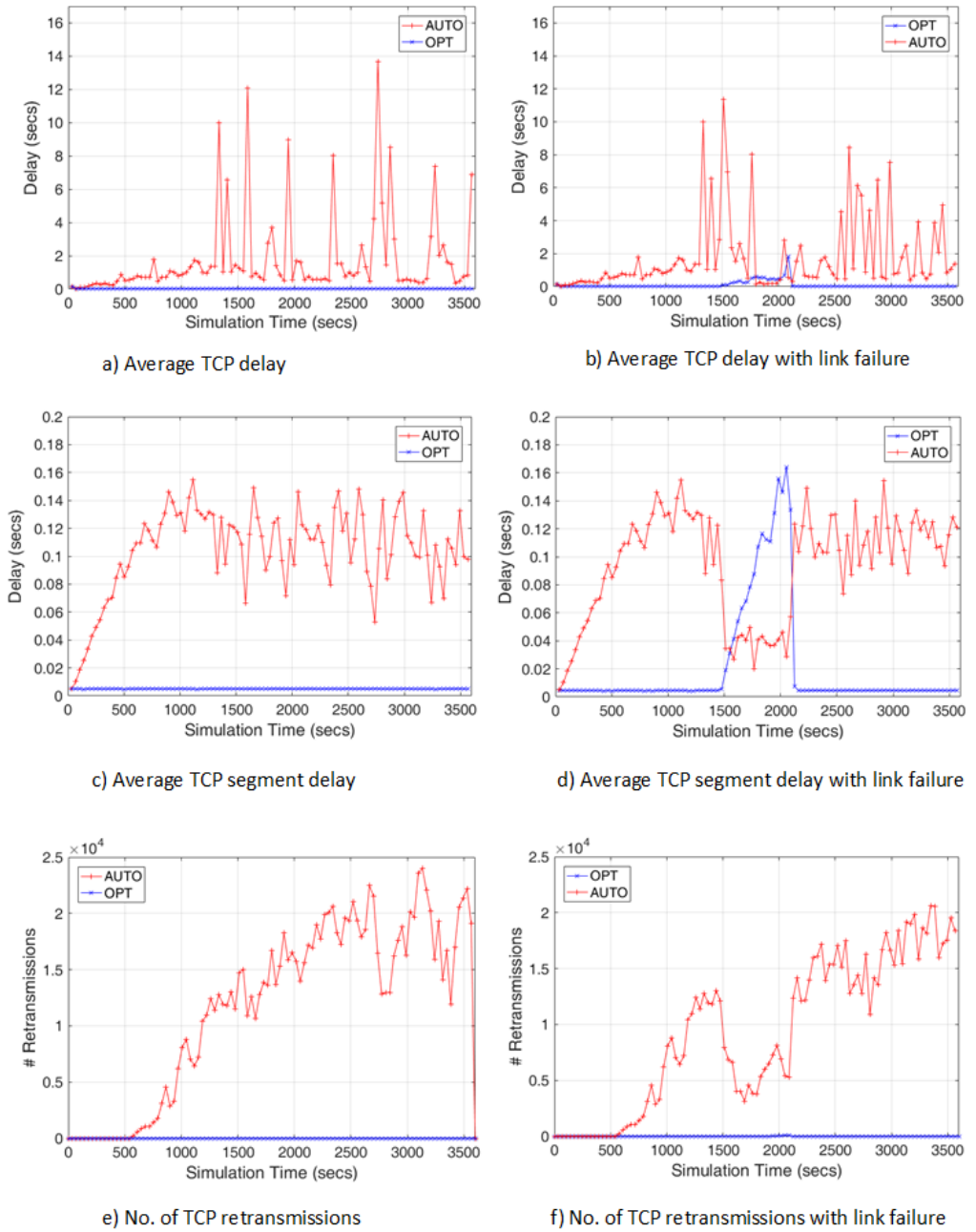


Figure 6.2: TCP Performance

TCP Delay

Figure 6.2a shows the average TCP delay under normal condition. We notice quite stable and low values in the OPT topology, compared to variable and at times, much higher values in the AUTO topology. This can be attributed to congestion on some of the major links in the AUTO topology. A maximum difference of 13.63 seconds is recorded at 2736 seconds into the 3600 seconds simulation. The delay in the OPT topology at this time is only 21 milliseconds.

Link failure causes the average TCP delay in the OPT topology to increase slightly, as can be seen in Figure 6.2b. However, after restoration, the values drop back to their normal millisecond range. On the contrary, the average TCP delay in the AUTO topology drops as a result of the failure, then rises again after restoration. This is an indication of a poor distribution of traffic, such that, when the link fails and new routes are calculated, under-utilized links are also exploited, leading to better (lower) delays.

TCP Segment Delay

TCP breaks downs large application data packets into multiple segments suitable for transport across the network, before sending them to their destination nodes. The average TCP segment delay is the mean delay (in seconds) of segments received by the TCP layer in all nodes and for all connections in the topology. It is measured from the time a TCP segment is sent from the source TCP layer to the time it is received by the TCP layer in the destination node.

With TCP segment delay, the effect of congestion and the difference between AUTO routing and OPT routing becomes more pronounced, as can be seen in Figure 6.2c and Figure 6.2d, for the normal and link failure scenarios, respectively.

TCP Retransmissions

TCP retransmission counts the number of times a TCP-segment have had to be resent because a previous copy did not arrive at its final destination. The number of TCP retransmissions is a good indicator of congestion and subsequent packet-drops in the network. When a packet is dropped as a result of congestion, TCP reduces its speed of transmission, then attempts to resend the dropped packet.

Figure 6.2e shows no retransmissions in the OPT routing topology during normal operations. In the AUTO routing topology, all starts well, with zero retransmissions in the first 500 seconds. Thereafter, the number of retransmissions virtually increases until the end of the simulation. A peak of 24008 retransmissions is reached at 3132 seconds into the 3600-seconds simulation.

When the link failure occurs, Figure 6.2f shows that the number of retransmission in the OPT routing topology, increases only slightly, from zero to a maximum of 82. It then falls back to zero immediately after the link is restored. This is negligible and does not adversely affect the TCP performance. In contrast, the link failure causes the already high retransmission rate in the AUTO routing topology to suddenly drop. But then, the number of retransmissions starts increasing again shortly after and even before the failed link is restored. A sharper increase is also observed immediately after restoration, which virtually increases all through until the end of the simulation run.

The observed increase in the number of retransmissions with time in the AUTO routing topology and the recorded drop in this number during failure, are further indicators and confirmation of congestion and poor traffic distribution in that topology. The good results in the OPT routing topology additionally confirm the advantages of optimization and the effectiveness of the traffic redistribution.

6.4.4 FTP Performance

We next look at the FTP performance, with respect to the amount of traffic sent and received, as well as to the recorded download response time.

Figure 6.3a shows the time-average of the FTP traffic that is sent in the AUTO routing and the OPT routing topologies during normal conditions. With a few exception in the first 500 seconds of the simulation, where the average FTP traffic sent (in bytes/s) by FTP servers to requesting FTP clients, appear to be slightly higher in the OPT routing topology than in the AUTO routing topology, the amount sent thereafter is approximately the same in both topologies. When failure occurs, Figure 6.3b shows only minor (AUTO routing topology) to negligible (OPT routing topology) changes in the scenarios respectively.

The FTP traffic received during normal and failure conditions respectively, show clear difference between the amount received in the AUTO routing topology and that received in the OPT routing topology. In the AUTO routing topology much less traffic is received than was sent (Figure 6.3c) and the impact of link failure in the topology is also clearly affects the amount being receive at that time (Figure 6.3d). We observe a slight increase in received traffic in the AUTO routing topology, resulting from the link failure. On the contrary, Figures 6.3c and 6.3d respectively also show that the amount of FTP traffic received in the OPT routing topology is practically the same amount that was sent and this amount is also not impacted by the link failure.

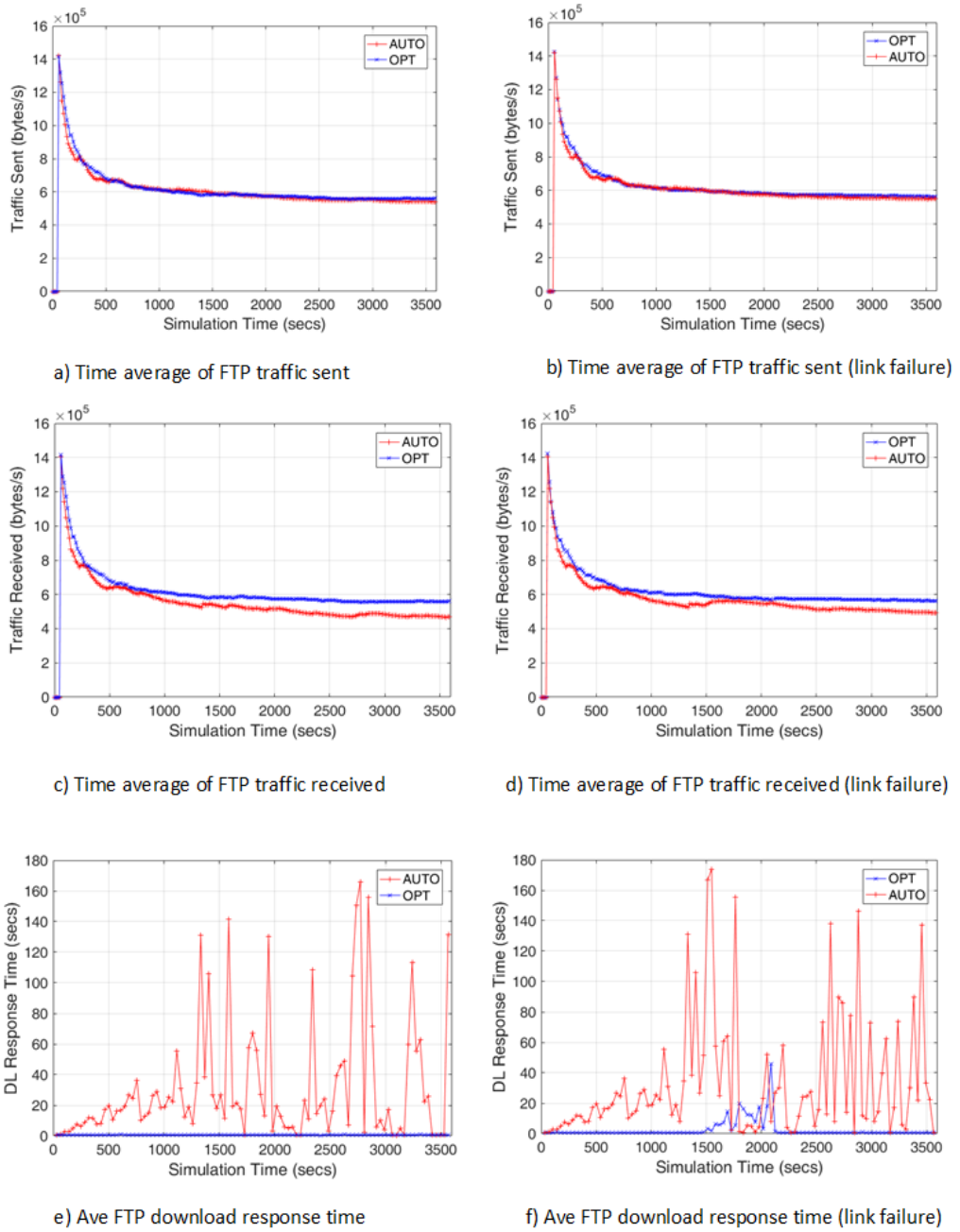


Figure 6.3: FTP performance

The average FTP download response times in the AUTO and OPT routing topologies are shown in Figure 6.3e, for normal operation and Figure 6.3f in the presence of link failure. During normal operation, we observe a very stable and low download response time in the OPT routing topology. On the contrary, the average download

response times in the AUTO routing topology increase right from the start and is quite dynamic. The values vary with time, reaching a maximum of 165.9 seconds. This is a great contrast to the maximum of only 1.04 seconds measured in the OPT routing topology.

Figure 6.3f shows interesting traits of the recorded FTP download response time, resulting from link failure. During the failure period, the download response time in the AUTO routing topology remains high at first, then falls to very low values before rising again after the link is restored. The maximum (worst) value of 173.85 seconds is recorded shortly after the failure, at 1548 seconds into the experiment. In the OPT routing topology, the failure also causes a rise in its download response time, to a maximum value of 45.65 seconds at 2088 seconds into the experiment. After the link is restored, the value drops back down to hundreds of milliseconds, therefore remaining below 1 second till the end of the simulation.

6.4.5 HTTP Performance

The average amount of HTTP traffic that is sent and received in each topology, are shown in Figure 6.4a to 6.4d, for the normal and link failure conditions respectively. The average amount of HTTP traffic sent (in bytes/s) during normal operations, is higher in the OPT routing topology than in the AUTO routing topology, as can be seen in Figure 6.3a. While the trend is virtually constant with time in the OPT routing topology, in the AUTO routing topology, it virtually slowly decreases with time. However, and unlike what we observed with FTP traffic in Section 6.4.4, the amount of HTTP traffic that was respectively sent in the OPT and AUTO routing topologies, is the same amount that is also respectively received, as shown in Figure 6.4c.

In the presence of a link failure, the same behavior is noticed for the HTTP traffic sent, Figure 6.4b as for the HTTP traffic received, Figure 6.4d. The amount of HTTP traffic in the AUTO routing topology rises immediately after link failure and remains high during the whole failure period. Immediately after the link is restored, it falls back to lower values. The amount of HTTP traffic in the OPT routing topology drops immediately after the link failure occurs and keeps dropping slowly until the link is restored. Thereafter it rises back to its old level and remains high until the end of the simulation.

The fact that the amount of HTTP traffic *sent* in the OPT and AUTO routing topologies differs by so much, Figure 6.4a, while the amount of FTP traffic *sent* shows no such difference between the two topologies, Figure 6.3a, only helps to reinforce the notion that different applications and protocols can be impacted differently by changes that are made on the network or to the traffic flowing through it. Therefore, careful planning, testing and cautious implementation are very necessary.

The average HTTP object response time is shown in Figure 6.4e for the AUTO and OPT routing topologies respectively. We notice that the average response time in the OPT routing topology remains constantly low, below 13 milliseconds (very good) from the start till the end of the simulation.

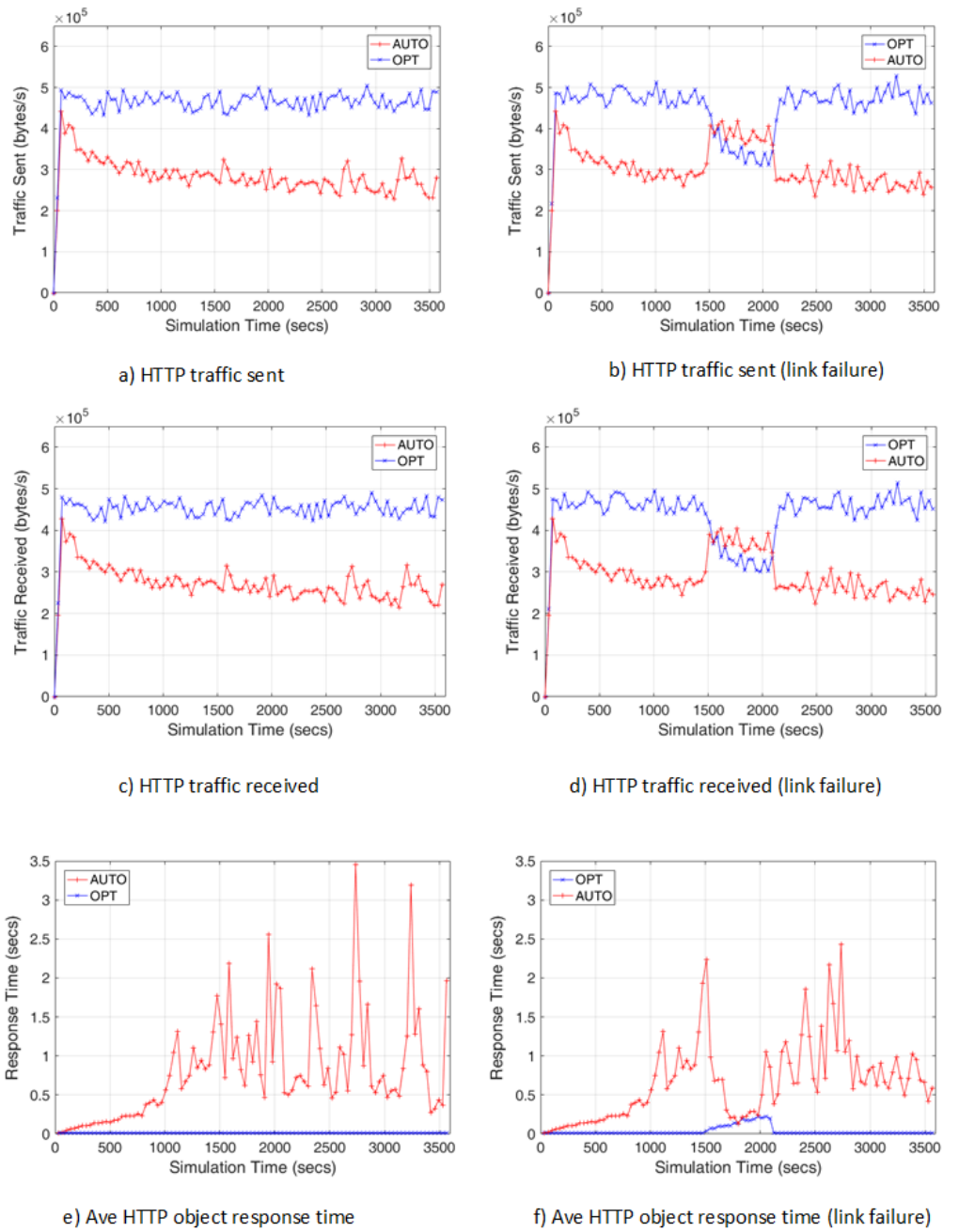


Figure 6.4: HTTP performance

In contrast, the average response time in the AUTO routing topology, slowly and steadily increases right from the start, until about 1080 seconds into the simulation. It becomes highly volatile from then on, while still increasing in trend. The maximum average response time of 3.45 seconds is recorded at 2736 seconds into the 3600-seconds simulation. This is more than 265 times higher than the highest recorded value in the OPT routing topology.

A link failure in the AUTO routing topology, causes a drop in HTTP average response time from a peak of 2.23 seconds to a trough of 131 milliseconds, as can be seen in Figure 6.4f. After the link is restored, the average response time increases again, reaching a peak of 2.43 seconds (the maximum in this scenario), before dropping to values at/below 1 second, towards the end of the simulation. We make a remarkable observation at this point, based on the recorded response time, which drops substantially immediately after the link failure occurs. After the link is restored, it does not only continue to rise, but later drops again to relatively lower values, as from the 2722 seconds time-point until the end of the simulation. The maximum response time recorded in this scenario is far below that recorded when there was no link failure.

In the OPT routing topology, the same link failure causes a minor increase in the average HTTP object response time. This slowly rises with the duration of the failure, from 13 milliseconds to 222 milliseconds at the time of restoration. However, immediately after the failed link is restored, the values drop back to their normal levels at/below 13 milliseconds.

6.4.6 Voice Performance

All the applications and protocols we've analyzed so far in this chapter are TCP-based. In this and the next sub-section, we are going to analyze the impact on UDP-based applications, i.e. impact on voice and video respectively.

We next analyze the amount of voice traffic sent when using AUTO and OPT routing, respectively. Figure 6.5a shows that slightly more voice traffic is sent in the OPT routing topology than in the AUTO routing topology. Figure 6.5b shows that there are no adverse effects on the voice traffic that is sent in both topologies, resulting from the link failure in each of the topologies.

Figure 6.5c shows the voice traffic that is received. It reveals that much less traffic is received in the AUTO routing topology than was originally sent. It also reveals that the amount of traffic received in the OPT topology, is the same as the amount that was originally sent. The effect of link failure on the amount of traffic that is received, is shown in Figure 6.5d. We can deduct from it that the amount of voice traffic that is received in the OPT routing topology is not affected by the failure. However, in the AUTO routing topology, we observe a slight increase in the received voice traffic during the failure period.

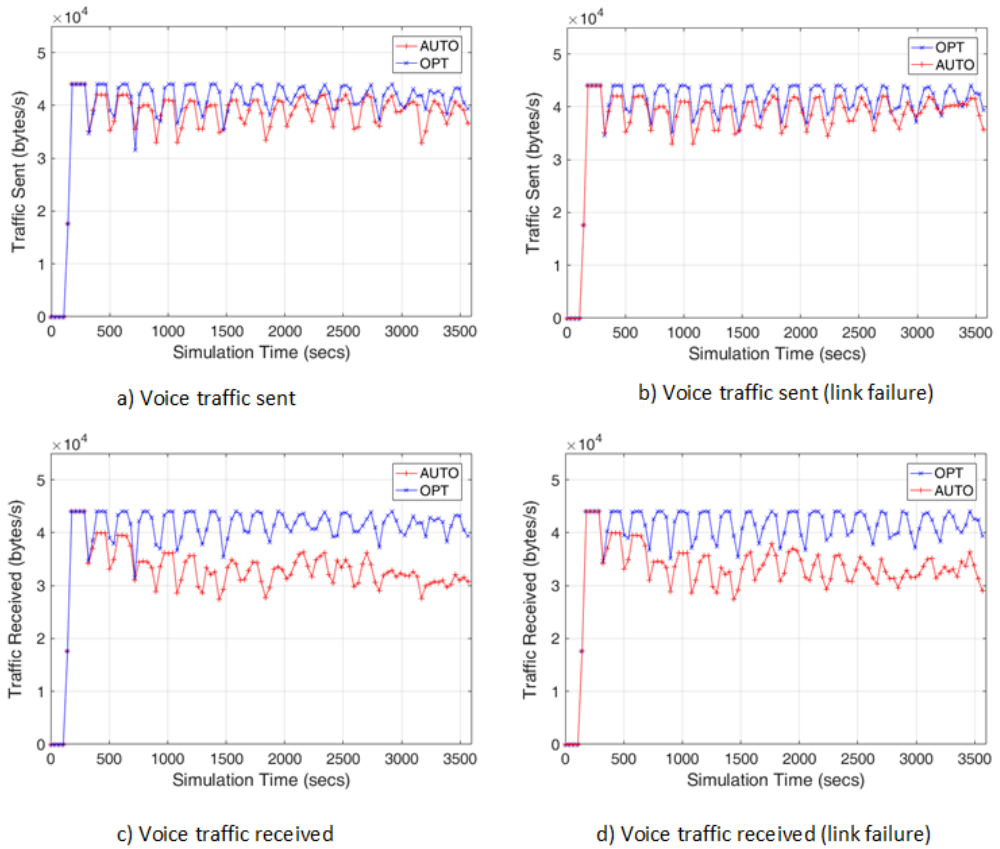


Figure 6.5: Voice sent and received traffic

Voice Packet Delay

The end-to-end delay of a voice packet, also known as "analog-to-analog" or "mouth-to-ear" delay, is the sum of the individual delays of many different components, the total of which is given by the following formula:

$$D_{Voice} = D_{net} + D_{enc} + D_{dec} + D_{compr} + D_{decompr}$$

where:

- D_{net} denotes the *network delay*, i.e. the time interval between when the sender node gave the packet to RTP to the time the receiver got it from RTP.
- D_{enc} denotes the *encoding delay* (on the sender node) and is computed from the encoder scheme.

- D_{dec} denotes the *decoding delay* (on the receiver node), which is assumed to be equal to the encoding delay.
- D_{compr} and $D_{decompr}$ denote the *compression* and *decompression delays* respectively and come from their corresponding attributes in the Voice application configuration of the simulator.

Th above values are automatically calculated by the simulator. Statistics from all active voice nodes in the network are collected and the average is calculated by the simulator. The International Telecommunication Union (ITU) G.114 standard recommends a maximum one-way delay of 150 milliseconds for voice packet transmissions [115].

The voice packet end-to-end delay under normal operation is shown in Figure 6.6a. We notice that, while the average delay in the OPT topology remains relatively constant from start to finish, that in the AUTO routing topology rises from the start and becomes quite dynamic with time. Its maximum value of 486.5 milliseconds is recorded at 3456 seconds of simulation time, while that in the OPT routing topology remains below 66 milliseconds.

Figure 6.6b shows the effect of link failure on voice packet end-to-end delay. While the value drops in the AUTO routing topology, as a result of the link failure, we observe a steady increase in the OPT routing topology. Immediately after the link is restored, the average delay value in the AUTO routing topology increases again, while that in the OPT routing topology drops back to the level it had before the failure.

Voice Packet Delay Variation

The voice packet delay variation is simply the difference in the end-to-end delays of selected voice packet pairs within the same stream [46]. Remember that the end-to-end delay for a voice packet is measured from the time it is created to the time it is received.

Figure 6.6c represents the measured voice packet delay variation under normal condition. It shows a constantly low voice packet delay variation (in picoseconds order of magnitude), when using OPT routing. In contrast, a relatively higher and variable voice packet delay variation is recorded when using AUTO routing. A maximum value of 81 milliseconds is attained at 2880 seconds into the simulation.

As can be seen from Figure 6.6d, the single link failure causes no visible change to the volatile pattern already observed in the AUTO routing topology, in the absence of failure (see Figure 6.6c). However, a noticeable change is seen in the OPT routing topology. Its voice packet delay variation increases from picoseconds (virtually zero) to a maximum of 21.4 milliseconds, as a result of the link failure. This however drops back to its pre-failure picoseconds-level immediately after restoration and remains

consistently at that level until the end of the simulation. Long after restoration in the AUTO routing topology, i.e. after 3456 of simulation time, the delay variation peaks to a value of 84.6 milliseconds, before dropping back again to averagely lower values.

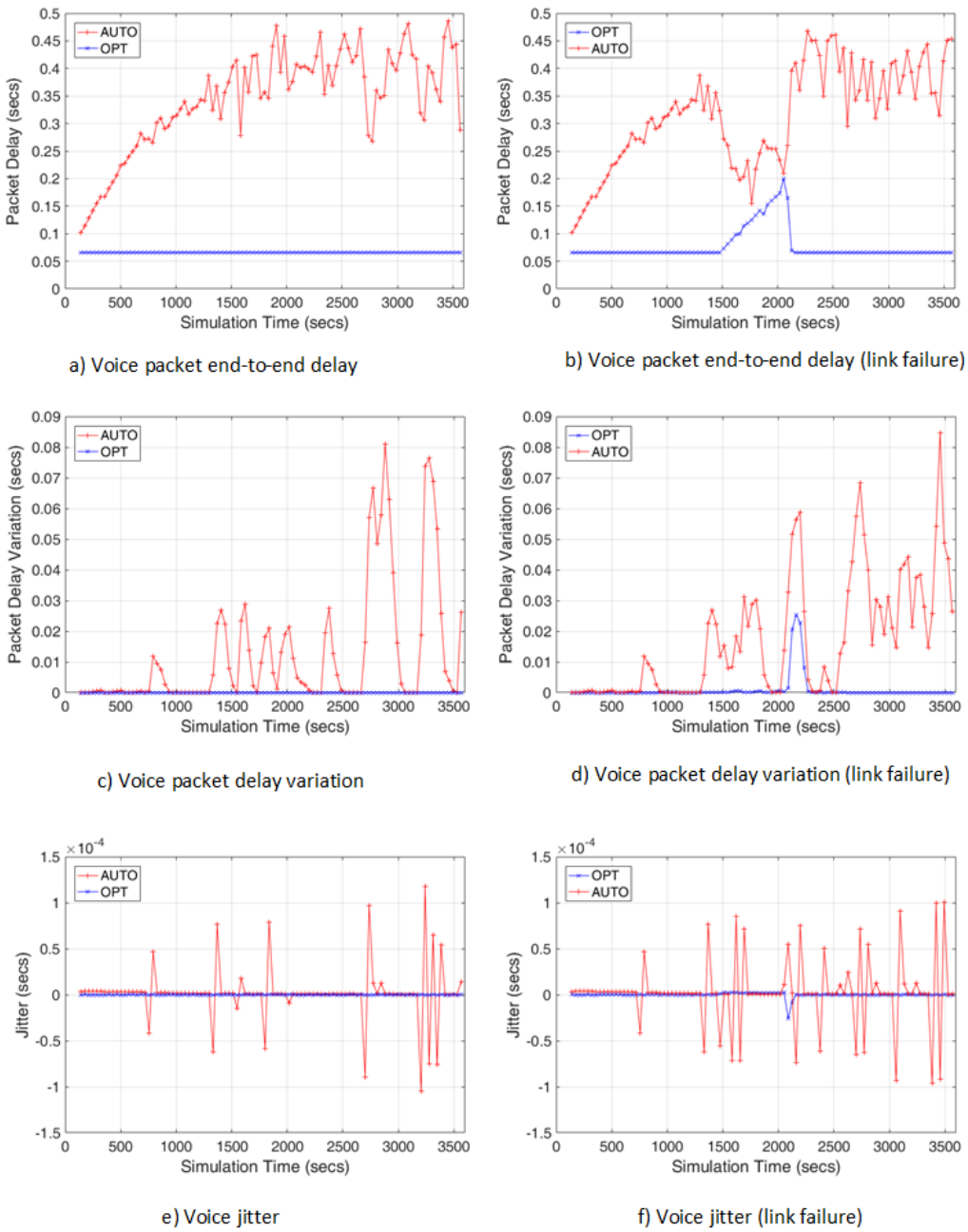


Figure 6.6: Voice performance

Voice Jitter

A more elaborate definition and description of voice jitter has already been provided in Chapter 5, Section 5.5.4. In this subsection, we will just present its values as measured in the AUTO and OPT topologies respectively.

Figure 6.6e shows the measured jitters in the AUTO and OPT routing topologies under normal conditions. The jitter in the OPT routing topology remains close to zero and flat across the full duration of the simulation, while that in the AUTO routing topology varies between positive and negative peaks at multiple time-points during the simulation.

In the OPT routing topology, the single link failure causes only a single minor change in jitter. In the AUTO routing topology, the same adds additional positive and negative peaks, as can be seen in Figure 6.6f.

Based on the measured jitter values and for the sake of comparison, we can conclude that the OPT routing in 20-Links topology offers better voice performance than AUTO routing in the same topology. We however also like to mention here that, despite this difference in jitter values, none of the measured jitters in either the AUTO routing topology or the OPT routing topology, comes even close to the accepted tolerance of 30 milliseconds.

6.4.7 Video Performance

Video streaming is the other UDP-based application that we use for performance analysis and comparison in this study. In this sub-section, we shall analyze its performance in the 20-Link topology, when AUTO and OPT routing are respectively used.

Sent and Received Video Packets

Figure 6.7a shows the amount of video traffic that is sent during normal operations. Figure 6.7b shows the mount that is sent when the link with the highest throughput fails. As can be seen from both graphs, the same amount of video traffic is respectively sent in the AUTO and OPT routing topologies, during normal operations, as well as in the presence of the single link failure. Not even the link failure seem to affect the amount of video traffic that is sent.

Looking at Figure 6.7c, i.e. the amount of video traffic that is received during normal operations, we notice a stark difference between values in the OPT routing topology and those in the AUTO routing topology. While all video traffic sent in the OPT routing topology are also received at their final destinations, in the AUTO routing

topology and after 555 seconds into the simulation, only a fraction of the sent traffic is received.

Even when the link with the highest throughput fails, Figure 6.7d shows that the amount of received video traffic in the OPT routing topology remains unaffected by the failure. In contrast, we notice a clear increase in the amount of video traffic that is received in the AUTO routing topology, during the failure period. A clear drop in the amount of video traffic received is also observed, immediately after the link is restored.

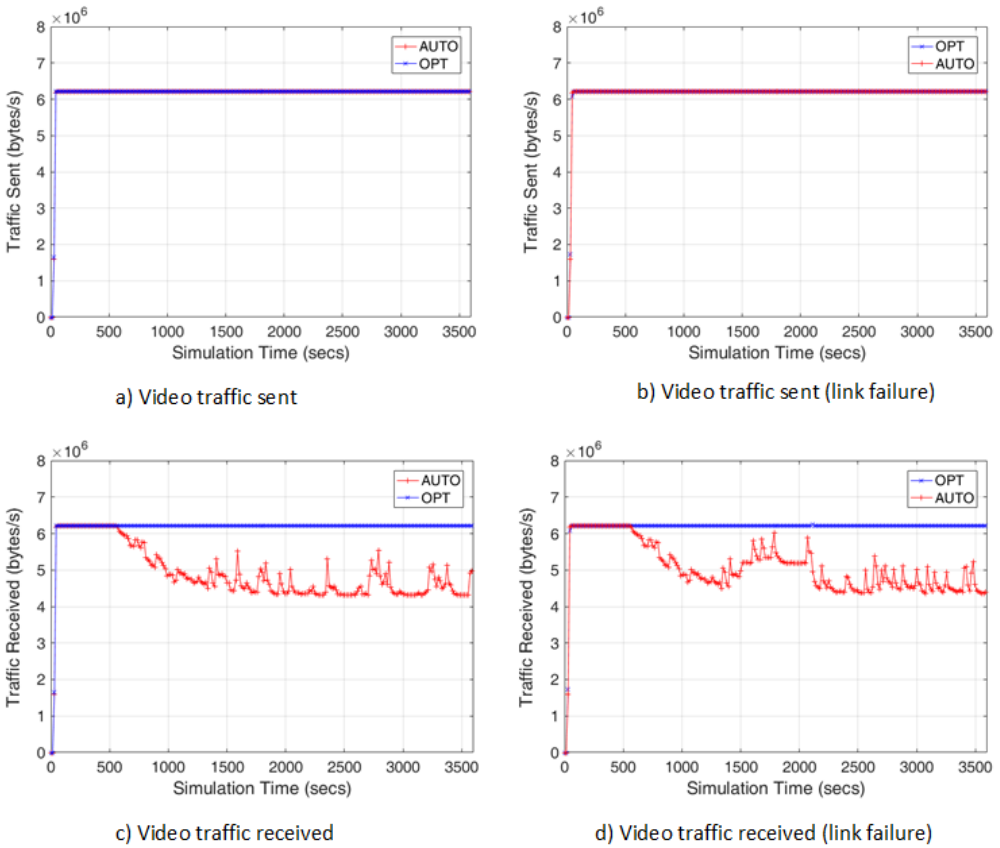


Figure 6.7: Video sent and received traffic

Video Packet Delay

Figure 6.8a shows the average video packet end-to-end delay during normal operation. The packet end-to-end delay in the OPT routing topology is shown to stay below 6 milliseconds from start to finish. In contrast to that, the average video packet end-to-end delay in the AUTO routing topology, is shown to increase steeply to a peak of 153 milliseconds within the first 612 seconds of the simulation. The

maximum value of 165.6 milliseconds is however recorded a few seconds later at the 720 seconds simulation time-point. A downward trend is then observed from thence until the end of the simulation, however, with momentary peaks in-between.

When the single link failure occurs, Figure 6.8b shows that the packet end-to-end delay in the AUTO routing topology, first decreases, then increases shortly before dropping again. However, immediately after the failed link is restored, it immediately rises again. In the OPT routing topology, the link failure causes the end-to-end delay to steadily increase during the full duration of the failure. Immediately after the link is restored, the delay drops back to the same low levels it had before the failure.

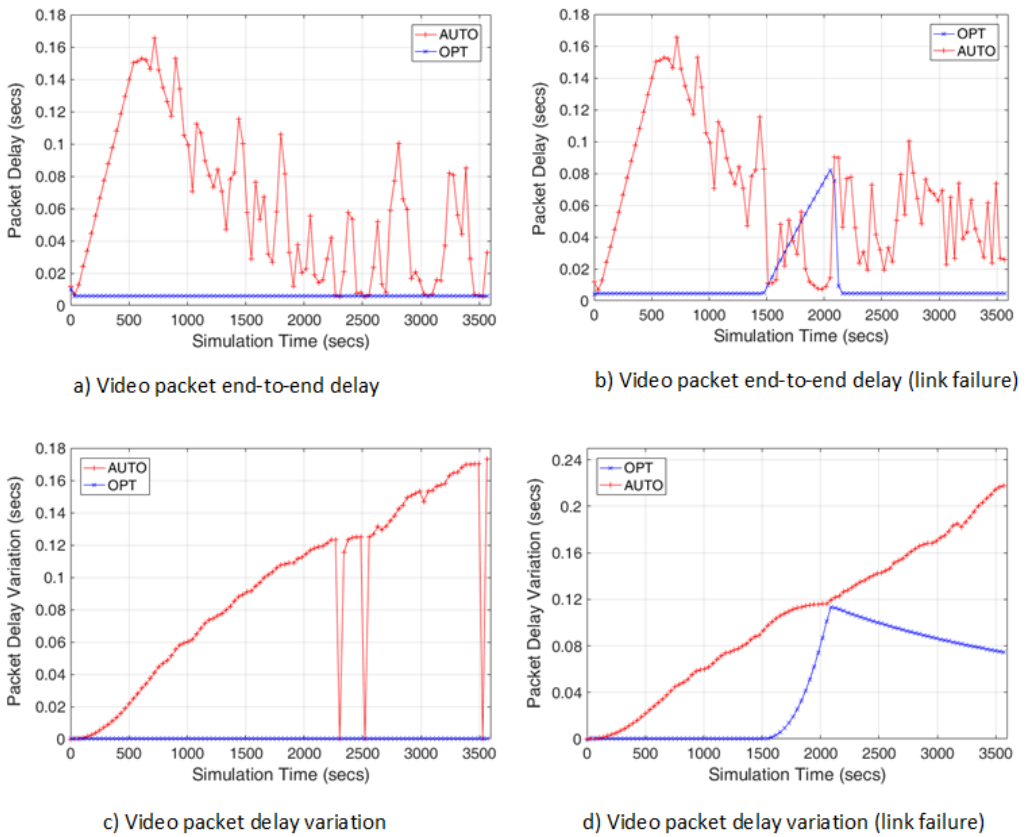


Figure 6.8: Video performance

Video Packet Delay Variation

The video packet delay variation under normal operating condition is shown in Figure 6.8c. We observe a constantly increasing packet delay variation in the AUTO

routing topology, while the same remains steadily low and close to zero, in the OPT routing topology.

Although the same rising pattern is also observed when link failure occurs in the AUTO routing topology, Figure 6.8d shows that the graph flattens a bit during the full period of the failure. It then rises again like before, after the failed link is restored. This means the link failure causes a slight reduction in the packet delay variation in the AUTO routing topology.

On the other hand, we notice that the packet delay variation in the OPT routing topology remains low as long as there are no failures. Immediately after the link fails, the value of the delay variation rises sharply, until the link is restored. Immediately after the restoration, it starts dropping, but not back to old levels. We instead observe a much slower decrease with time. As Figure 6.8d shows, despite the steady drop from then until the end of the simulation, the value still remains relatively higher than what it was before the failure occurred.

6.5 Summary

To improve the performance in a congested backbone network, we first formulate and solve a Mixed-Integer Programming (MIP) problem based on the same topology. The solution provides optimized OSPF interface costs, which we manually transfer to our main simulator. We next design two kinds of topology, based on the same 20-Links topology, but using two variants of OSPF routing. In one (the AUTO routing topology), OSPF uses the default (automatic) calculated routing metrics. In the other (the OPT routing topology), it uses the MIP-determined and manually added optimized metrics. We run two sets of simulations for each routing variant, one without link failure and the other with a 10-minute single link failure in the middle of the simulation.

The obtained results show improved performance of many magnitudes in scenarios based on OPT routing. However, in some failure instances, we notice better performance in the AUTO routing topology against that in the OPT routing topology. The reasons for this are provided in the subsequent paragraphs.

Normally, when a link fails, it triggers the recalculation and selection of alternative paths that do not include the failed link. However, if the failed link belongs to a set of equal-cost paths, OSPF simply continues to push traffic via the remaining accessible paths, without generating new routing updates.

First reason: We notice that the AUTO routing topology has no equal-cost paths, while the OPT routing topology has multiple equal-cost paths for multiple Origin-Destination (OD) pairs. Further analysis reveals that the failed link belongs to one of such. Therefore, while OSPF is able to re-calculate new costs and determine new paths for traffic flows in the AUTO routing topology that are affected by the link

failure, in the OPT routing topology, it simply reroutes the affected flows via the remaining paths in the equal-cost set.

Second reason: In the OPT routing topology, equal-cost does not necessarily mean equal bandwidth/capacity. Therefore, in the special case where link failure forces traffic redistribution via paths with lower capacity, these might immediately suffer under the additional load, if more capacity is required than is available on the links (or paths).

The second reason above also explains why Table 6.2b that records utilization and throughput in the OPT routing topology during failure, has a few "red" (over-utilized) links, but Table 6.1b that records the same under normal (no failure) condition, has none.

7 Conclusion

This chapter summarizes our findings, discusses lessons learned and outlines future research directions.

Internet traffic is expected to keep growing, as more users, devices, services and applications are being added. With the packet forwarding Internet also taking over more fundamental roles, e.g. from legacy infrastructures such as circuit-switched telecommunication and radio/TV broadcast communications, its traffic composition is also expected to increase and exhibit much broader diversity.

A detailed understanding of modern Internet traffic composition and behavioral patterns is necessary to properly address their associated management and performance challenges.

7.1 Managing P2P Traffic

Content-sharing P2P networks build logical (application-level) topologies on top of the Internet's routing underlay topology. The large proportion of backbone capacity they consume, the disruptive nature of their traffic and the huge challenges these all pose to ISPs, provoked huge interest in the research community, leading to multiple research studies and better understanding of P2P systems and the issues they create.

In this thesis, we propose a novel, practical and simple approach that, on one hand, helps ISPs address the traffic management and performance challenges attributed to P2P traffic and on the other hand, boosts general P2P user experience. This win-win solution is attained via ISP and P2P collaboration, enabled by our newly proposed service, which we call, the *Oracle* service.

To assess our propositions and quantify the gains made possible by using the Oracle service, we carry out a series of packet-level simulations on unbiased and oracle-biased topologies. We then analyze and compare their performance using selected metrics. Multiple benefits are observed in Oracle-biased topologies.

First, we analyze the graph structural properties of the unbiased and biased overlay networks. We use a visualization tool to show the stark structural contrast that exist between the unbiased and biased P2P overlays. While the unbiased overlay is shown to have no particular structure, the biased overlay is shown to have a structure that is more aligned with the ISP's underlay topology, since the Oracle service facilitates the construction of a more meaningful (localized) overlay. This mitigates the overlay/underlay mismatch, which in unbiased overlays, is caused by the random and selfish connection approach that peers use. Although biased overlays show a slight increase in graph diameter and small decrease in mean node degree, these are shown to have no adverse effects on the network performance as a whole.

We next analyze the distribution of queries and responses (hits) in the overlays, including the inter-AS traffic, then compare the measured average download times of peers in the unbiased and Oracle-biased topologies. We observe unhampered distribution of queries and responses in the Oracle-biased overlay, just like they are in the unbiased overlay. However, in the case of the biased overlay, most of these are localized within an AS. The ISP thus retains a huge proportion of the P2P traffic within its own domain and also reduces the proportion of inter-AS traffic by large factors. These result in huge cost savings for the ISP and faster content download times for the peers.

To evaluate the performance of the Oracle service under different user behavioral conditions, we carry out additional experiments using mathematical distributions that represent best, normal and worst conditions. In all but a few cases, the results show that performance in the Oracle-biased overlay are much better than in the unbiased overlay.

Based on the presented results, we argue that offering the Oracle service brings lots of benefits to the ISP. These include increase control over P2P traffic, improved traffic engineering ability, better service to customers and huge cost savings on transit fees. By using the Oracle service, P2P users also contribute to the increased locality and reap huge benefits from it as well. They no longer need to infer network conditions themselves. Increased locality of query hits translates into better download times for the peers.

In general, reduced overlay/underlay mismatch, increased P2P locality and reduction in overhead traffic, amount to a more efficient and scalable network.

7.2 Topologies and Traffic Flows

To study how the topology of a backbone network affects traffic flow and general performance, we model 3 distinct topologies that have the same number of nodes (i.e. 12), but differ in the number of their links, structural characteristics and total bandwidth capacity. The 3 topologies consists of i) a Fullmesh topology, with 66

links and the highest total bandwidth capacity of approximately 285 Gbps, ii) a 30-Links topology with a moderate total capacity of approximately 195 Gbps and iii) a 20-Links topology, with the least total capacity of approximately 132.5 Gbps.

We perform a series of simulation studies, applying the same traffic load and network conditions on each of the 3 topologies, then analyze and compare their effects on performance using selected performance metrics.

For each topology, we analyze the performance under normal (baseline) traffic load and when the traffic load increases by 35%. We also analyze the effect of a single link failure under baseline traffic, as well as when the traffic load increases by 35%. For the single link failure analysis, we select the link in each topology that has the highest throughput at baseline traffic.

In nearly all analyzed and compared categories, the best performances are observed in the Fullmesh topology, which also shows the best tolerance and resilience against the single link failure, even at increased traffic load. However, for large backbone networks with many nodes, full-mesh topologies are impractical because they lack scalability and are very expensive to build and operate. Despite the high path diversity in the Fullmesh topology, we observe that only a few of these paths are used. This can be attributed to the default best path route-selection mechanism of the routing protocol that selects only the best path (based on metrics) out of many potential paths to include in its forwarding table. As a result, many links in the topology are not used at all.

On the contrary, the 20-Links topology, with less than one-third the number of links and a little less than half the total capacity of the Fullmesh topology, shows comparable performance only a in a very limited number of cases, during baseline traffic load. In most cases, the performance in the 20-Links topology comes last. The same topology is also observed to have the least tolerance and resiliency against link failure.

In a majority of the cases, the 30-Links topology, which has less than half the number of links, but more than two-thirds the capacity of the Fullmesh topology, is observed to offer the same level of performance as the Fullmesh topology. However, it also shows less tolerance and resilience against link failure, compared to those offered in the Fullmesh topology. Nevertheless, one can still argue that, because of its enormously reduced number of links and comparable performance to that of the Fullmesh topology, it offers the best cost/performance/tolerance trade-off of all 3 topologies, while the 20-Links topology offers the least trade-off.

As a reference network topology used by many other research groups, the reference backbone network topology for Germany offers us the ability to obtain results that are generally verifiable and comparable. We use the 12 node model to reflect recent updates to the original model. The reduced number of nodes in the new reference topology also has additional advantages, such as reduced CPU processing and shorter simulation runs. We still do think that larger topologies consisting of many more

nodes would generally offer similar results if these share the same kind of structural properties (e.g. average node degree) and bandwidth distribution like the topologies used in our studies.

7.3 Optimizing Traffic Flows

On further assessment of the 20-Links topology under increased traffic load, an important observation is made. A few major links are noticed to be suffering from congestion, while others (of lesser bandwidth) have little or no traffic on them. Could this situation be improved if the flow is optimized to avoid congestion on such links? We think and show that it could be improved.

We propose an optimization method that exploits the use of MIP to minimize the maximum utilization on each link and efficiently distribute the traffic load in the topology. This approach offers many benefits to the ISP, since it does not involve the addition of new links/nodes or capacity upgrades on exiting links and can be accomplished within minutes. It is of major importance that the general performance must not suffer, as a result of the proposed approach. Performance analysis therefore constitutes a major part of the analyses to assess the practicability and usability of this approach.

MIP is used to determine the best interface cost for optimized OSPF routing within the topology. The performance in the optimized routing topology (OPT) and that in the automatic (default) routing topology (AUTO) are analyzed and compared.

The results show significant differences in performance between the two compared topologies. The OPT routing topology is observed to offer better performances than the AUTO routing topology in nearly all compared categories. Despite the better performance in the absence of link failure, the OPT topology shows less resilience to failure than the AUTO topology. The is however not a general issue, but one constraint to our selected topology. The reasons for this are given in Chapter 6 Section 6.5. The issue can easily be resolved or be prevented through careful planning and analysis.

The cost-savings with this approach are enormous. Instead of immediately upgrading those links in the topology that appear to be congested, an ISP can use our proposed cost-effective, fast and straight-forward approach to engineer the traffic flow and so, postpone expensive upgrades by a few cycles into the future.

Generally, underutilized or sufficiently over-provisioned networks benefit little from (and do not need) such optimization, since the routing protocol (e.g. OSPF and IS-IS, via use of the SPF algorithm) already ensures that the best route is selected for packet forwarding. The Fullmesh and 30-Links topologies fall in this category. However, in networks where the capacity on some paths start to run out *and* the network still possesses enough capacity on other paths, optimization becomes a

viable and less expensive option. The 20-Links topology (at 35% increased traffic) falls in this category.

7.4 Outlook

We have shown in this thesis how the ISP can use collaboration to address fundamental traffic management challenges on its IP backbone network. We have also shown how in the absence of such collaboration, the ISP can still use other already available resources to control and manage traffic flows. The challenges posed by P2P can also be posed by any other future application or trend. Our work on ISP and P2P collaboration offers a foundational approach and example for similar cases of joint interest, involving even competing parties. It shows how collaboration facilitates mitigation of such issues and offers benefits to all parties. The Internet community recognizes the need for such and is currently working on standardized protocols based on combined contributions from our work and those from a few other groups.

The ISP's readiness to deal with highly dynamic conditions and fast-paced activities on the Internet, demands for innovative approaches that do not only offer solutions, but for approaches that offer quick and cost-effective solutions. With regards to traffic management, forecasts and trends (e.g. Artificial Intelligence (AI), Internet of Things (IoT), big data and cloud services) indicate increasingly that more complex challenges await the ISP in the near future.

Irrespective of all past and current traffic management and flow optimization approaches, the one thing that most researchers and operators agree on, is that a radical change of the network's current architecture is necessary to effectively deal with future traffic compositions/volumes/flows and their management challenges [12, 157, 174]. The good news is that this change has already begun. With the recent shift towards Software Defined Networking (SDN) [11, 116, 127, 140, 153], the virtualization of the network [29, 30, 48, 222] and the separation of the control, data and management planes into separate entities, the network infrastructure is becoming more simplified. It can now benefit from better programmability, scalability and flexibility. This is giving rise to newer architectures, such as the segment routing architecture [31, 58, 60], which is much more scalable, flexible, less complex and easier to manage/operate than IP routing or MPLS. Further, with easier accessibility to the most up-to-date information on network conditions, controllers in SDNs are now able to react much faster to changes and failures. This enormously facilitates traffic engineering in SDNs, far beyond the levels possible in current and past networks.

Appropriate measurement approaches will always accompany whatever architectural changes are made, in order for ISPs to be able to assess and fully understand the impact of these combined effects on their network. For example, the current big

data revolution demands for appropriate measurement and handling tools by their respective stakeholders. This includes ISPs, whose networks provide the medium through which big data flows from the different sources where they are generated to the respective locations where they are collected/analyzed.

List of Figures

List of Tables

Bibliography

[1] Eytan Adar and Bernardo A Huberman. "Free riding on Gnutella". In: *First monday* 5.10 (2000).

[2] Bernhard Ager, Nikolaos Chatzis, Anja Feldmann, Nadi Sarrar, Steve Uhlig, and Walter Willinger. "Anatomy of a large European IXP". In: *ACM SIGCOMM Computer Communication Review* 42.4 (2012), pp. 163–174.

[3] Vinay Aggarwal, Obi Akonjang, and Anja Feldmann. "Improving User and ISP Experience through ISP-aided P2P Locality". In: *Proceedings of 11th IEEE Global Internet Symposium 2008 (GI '08)* (2008).

[4] Vinay Aggarwal, Anja Feldmann, and Christian Scheideler. "Can ISPs and P2P users cooperate for improved performance?" In: *ACM SIGCOMM Computer Communication Review* 37.3 (2007), pp. 29–40.

[5] Vinay Aggarwal, Anja Feldmann, Marco Gaertler, Robert Görke, and Dorothea Wagner. "Analysis of Overlay-Underlay Topology Correlation using Visualization". In: (2006). Outstanding Paper Award, 385–392. URL: http://www.net.t-labs.tu-berlin.de/papers/AFGGW-AOTCV-05a.pdf.

[6] Ravindra K Ahuja. *Network Flows: Theory, Algorithms, and Applications*. Pearson Education, 2017.

[7] O Akonjang, A Feldmann, S Previdi, B Davie, and D Saucez. "The PROXIDOR service". In: *IETF Draft, Mar* 2 (2009).

[8] Obi Akonjang, Vinay Aggarwal, Anja Feldmann, Jun Jiang, and Pengchun Xie. *The Oracle Protocol*. Draft v01. 2008. URL: http://www.net.t-labs.tu-berlin.de/papers/AAFJP-TOPDV1-08.pdf.

[9] Alexa. *The top 500 sites on the web*. URL: https://www.alexa.com/topsites.

[10] Richard Alimi, R Penno, Y Yang, S Kiesel, S Previdi, W Roome, S Shalunov, and R Woundy. *Application-layer traffic optimization (ALTO) protocol*. Tech. rep. 2014.

[11] Rodolfo Alvizu, Guido Maier, Navin Kukreja, Achille Pattavina, Roberto Morro, Alessandro Capello, and Carlo Cavazzoni. "Comprehensive survey on T-SDN: Software-defined networking for transport networks". In: *IEEE Communications Surveys & Tutorials* 19.4 (2017), pp. 2232–2283.

[12] Thomas Anderson, Larry Peterson, Scott Shenker, and Jonathan Turner. "Overcoming the Internet Impasse Through Virtualization". In: *Computer* 38.4 (2005), pp. 34–41.

[13] Daniel Awduche, Angela Chiu, Anwar Elwalid, Indra Widjaja, and XiPeng Xiao. *Overview and principles of Internet traffic engineering*. Tech. rep. 2002.

[14] Daniel O Awduche and Johnson Agogbua. “Requirements for traffic engineering over MPLS”. In: (1999).

[15] Paul Barford and Mark Crovella. “Generating Representative Web Workloads for Network and Server Performance Evaluation”. In: *ACM SIGMETRICS Performance Evaluation Review*. Vol. 26. 1. ACM. 1998, pp. 151–160.

[16] Timothy J Berners-Lee and Robert Cailliau. “World-wide web”. In: (1992).

[17] Andreas Betker, Christoph Gerlach, Ralf Hülsermann, Monika Jäger, Marc Barry, Stefan Bodamer, Jan Späth, Christoph Gauger, and Martin Köhn. “Reference Transport Network Scenarios”. In: *MultiTeraNet Report* (2003).

[18] Richard Blum. *Network performance open source toolkit: using Netperf, tcptrace, NISTnet, and SSFNet*. John Wiley & Sons, 2003.

[19] Olivier Bonaventure. *Computer Networking: Principles, Protocols, and Practice*. The Saylor Foundation, 2011.

[20] R. Bonica, R. Thomas, J. Linkova, C. Lenart, and M. Boucadair. *PROBE: A Utility for Probing Interfaces*. RFC 8335. RFC Editor, 2018. URL: http://www.rfc-editor.org/rfc/rfc8335.txt.

[21] John Buford, Heather Yu, and Eng Keong Lua. *P2P networking and applications*. Morgan Kaufmann, 2009.

[22] Miguel Castro, Peter Druschel, Y Charlie Hu, and Antony Rowstron. *Proximity neighbor selection in tree-based structured peer-to-peer overlays*. Tech. rep. technical report MSR-TR-2003-52, Microsoft Research, 2003.

[23] Hyunseok Chang, Sugih Jamin, and Walter Willinger. “Inferring AS-level Internet topology from router-level path traces”. In: *Scalability and traffic control in IP networks*. Vol. 4526. International Society for Optics and Photonics. 2001, pp. 196–208.

[24] Nikolaos Chatzis, Georgios Smaragdakis, and Anja Feldmann. “On the importance of Internet eXchange Points for today’s Internet ecosystem”. In: *arXiv preprint arXiv:1307.5264* (2013).

[25] Nikolaos Chatzis, Georgios Smaragdakis, Jan Böttger, Thomas Krenc, and Anja Feldmann. “On the benefits of using a large IXP as an Internet vantage point”. In: *Proceedings of the 2013 conference on Internet measurement conference*. ACM. 2013, pp. 333–346.

[26] Der-San Chen, Robert G Batson, and Yu Dang. *Applied Integer Programming: Modeling and Solution*. John Wiley & Sons, 2011.

[27] Marco Chiesa, Guy Kindler, and Michael Schapira. “Traffic Engineering with Equal-Cost-Multipath: An algorithmic perspective”. In: *IEEE/ACM Transactions on Networking (TON)* 25.2 (2017), pp. 779–792.

[28] John W Chinneck. “Practical Optimization: A Gentle Introduction”. In: *Systems and Computer Engineering), Carleton University, Ottawa.* (2016). URL: http://www.sce.carleton.ca/faculty/chinneck/po.html.

[29] NM Mosharaf Kabir Chowdhury and Raouf Boutaba. "A survey of network virtualization". In: *Computer Networks* 54.5 (2010), pp. 862–876.

[30] NM Mosharaf Kabir Chowdhury and Raouf Boutaba. "Network virtualization: state of the art and research challenges". In: *IEEE Communications magazine* 47.7 (2009).

[31] Antonio Cianfrani, Marco Listanti, and Marco Polverini. "Incremental Deployment of Segment Routing Into an ISP Network: a Traffic Engineering Perspective". In: *IEEE/ACM Transactions on Networking* 25.5 (2017), pp. 3146–3160.

[32] Cecilia Cid, Marc Ruiz, Luis Velasco, and Gabriel Junyent. "Costs and Revenues Models for Optical Networks Architectures Comparison". In: *In Proceedings of IX Workshop in GMPLS Networks, Girona July*. Vol. 2010. 2010.

[33] CIDR. *CIDR REPORT for 22 Jan 18*. URL: https://www.cidr-report.org/as2.0/.

[34] B. Claise, B. Trammell, and P. Aitken. *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*. STD 77. RFC Editor, 2013. URL: http://www.rfc-editor.org/rfc/rfc7011.txt.

[35] Benoit Claise. *Cisco systems netflow services export version 9*. Tech. rep. 2004.

[36] cnet.com. *Comcast says it isn't throttling heavy internet users anymore*. 2018. URL: https://www.cnet.com/news/comcast-says-it-isnt-throttling-heavy-internet-users-anymore/.

[37] Bram Cohen. *The BitTorrent protocol specification, version 11031*. 2008.

[38] The Committee on Communications Policy. *Network Traffic Management and the Evolving Internet*. URL: https://ieeeusa.org/wp-content/uploads/2017/07/IEEEUSAWP-NTM2010.pdf.

[39] CORDIS. *Industry-Driven Elastic and Adaptive Lambda Infrastructure for Service and Transport Networks (IDEALIST)*. 2015. URL: http://cordis.europa.eu/.

[40] Renesys Corporation. *Scalable Simulator Framework (SSFNet)*. URL: http://www.ssfnet.org/homePage.html.

[41] IBM ILOG CPLEX. "12.7, User's Manual for CPLEX, 2016". In: *CPLEX division* (2016). URL: https://www.ibm.com/analytics/cplex-optimizer.

[42] Mark Crovella and Balachander Krishnamurthy. *Internet measurement: infrastructure, traffic and applications*. John Wiley & Sons, Inc., 2006.

[43] Mariela Curiel and Ana Pont. "Workload Generators for Web-Based Systems: Characteristics, Current Status, and Challenges". In: *IEEE Communications Surveys & Tutorials* 20.2 (2018), pp. 1526–1546.

[44] Saurav Das, Guru Parulkar, and Nick McKeown. “Rethinking IP core networks”. In: *Journal of Optical Communications and Networking* 5.12 (2013), pp. 1431–1442.

[45] Saurav Das, Guru Parulkar, Nick McKeown, Preeti Singh, Daniel Getachew, and Lyndon Ong. “Packet and Circuit Network Convergence with OpenFlow”. In: *Optical Fiber Communication Conference*. Optical Society of America. 2010, OTuG1.

[46] C. Demichelis and P. Chimento. *IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)*. RFC 3393. RFC Editor, 2002. URL: http://www.rfc-editor.org/rfc/rfc3393.txt.

[47] John Dilley, Bruce Maggs, Jay Parikh, Harald Prokop, Ramesh Sitaraman, and Bill Weihl. “Globally distributed content delivery”. In: *IEEE Internet Computing* 6.5 (2002), pp. 50–58.

[48] Dmitry Drutskoy, Eric Keller, and Jennifer Rexford. “Scalable network virtualization in software-defined networks”. In: *IEEE Internet Computing* 17.2 (2013), pp. 20–27.

[49] J. Fabini and A. Morton. *Advanced Stream and Sampling Framework for IP Performance Metrics (IPPM)*. RFC 7312. RFC Editor, 2014. URL: http://www.rfc-editor.org/rfc/rfc7312.txt.

[50] TK Fachbegriffe. URL: http://www.tk-fachbegriffe.de/index.php?id2=2400&a=320.

[51] Kevin R Fall and W Richard Stevens. *TCP/IP illustrated, volume 1: The protocols*. addison-Wesley, 2011.

[52] Michal Feldman and John Chuang. “Overcoming free-riding behavior in peer-to-peer systems”. In: *ACM sigecom exchanges* 5.4 (2005), pp. 41–50.

[53] Anja Feldmann, Albert Greenberg, Carsten Lund, Nick Reingold, Jennifer Rexford, and Fred True. “Deriving traffic demands for operational IP networks: Methodology and experience”. In: *IEEE/ACM Transactions on Networking (ToN)* 9.3 (2001), pp. 265–280.

[54] Anja Feldmann, Albert Greenberg, Carsten Lund, Nick Reingold, and Jennifer Rexford. “NetScope: Traffic engineering for IP networks”. In: *IEEE Network* 14.2 (2000), pp. 11–19.

[55] FICO. “FICO Xpress Optimization”. In: (2017). URL: https://www.fico.com/en/products/fico-xpress-optimization.

[56] Roy Fielding, Jim Gettys, Jeffrey Mogul, Henrik Frystyk, Larry Masinter, Paul Leach, and Tim Berners-Lee. *Hypertext transfer protocol–HTTP/1.1*. Tech. rep. 1999.

[57] Roy T. Fielding, James Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach, and Tim Berners-Lee. *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616. RFC Editor, 1999. URL: http://www.rfc-editor.org/rfc/rfc2616.txt.

[58] C. Filsfils, S. Previdi, L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir. *Segment Routing Architecture*. RFC 8402. 2018. URL: https://tools.ietf.org/html/rfc8402.

[59] Clarence Filsfils, Thomas Telkamp, and Paolo Lucente. "Best Practices in Network Planning and Traffic Engineering". In: *RIPE 61, Rome* (2011).

[60] Clarence Filsfils, Nagendra Kumar Nainar, Carlos Pignataro, Juan Camilo Cardona, and Pierre Francois. "The Segment Routing Architecture". In: *Global Communications Conference (GLOBECOM), 2015 IEEE*. IEEE. 2015, pp. 1–6.

[61] Simon Fischer, Nils Kammenhuber, and Anja Feldmann. "REPLEX: Dynamic Traffic Engineering based on Wardrop Routing Policies". In: *Proceedings of the 2006 ACM CoNEXT conference*. ACM. 2006, p. 1.

[62] Robert W Floyd. "Algorithm 97: shortest path". In: *Communications of the ACM* 5.6 (1962), p. 345.

[63] Sally Floyd and Vern Paxson. "Difficulties in simulating the Internet". In: *IEEE/ACM Transactions on Networking (ToN)* 9.4 (2001), pp. 392–403.

[64] P. Ford-Hutchinson. *Securing FTP with TLS*. RFC 4217. RFC Editor, 2005. URL: http://www.rfc-editor.org/rfc/rfc4217.txt.

[65] Behrouz A. Forouzan. *TCP/IP Protocol Suite (Mcgraw-hill Forouzan Networking)*. McGraw-Hill Education, 2009. ISBN: 0073376043.

[66] Bernard Fortz, Jennifer Rexford, and Mikkel Thorup. "Traffic Engineering with Traditional IP Routing Protocols". In: *IEEE communications Magazine* 40.10 (2002), pp. 118–124.

[67] Bernard Fortz and Mikkel Thorup. "Increasing internet capacity using local search". In: *Computational Optimization and Applications* 29.1 (2004), pp. 13–48.

[68] Bernard Fortz and Mikkel Thorup. "Internet traffic engineering by optimizing OSPF weights". In: *INFOCOM 2000. Nineteenth annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE*. Vol. 2. IEEE. 2000, pp. 519–528.

[69] Bernard Fortz and Mikkel Thorup. "Optimizing OSPF/IS-IS Weights in a Changing World". In: *IEEE journal on selected areas in communications* 20.4 (2002), pp. 756–767.

[70] Pierre Francois, Mike Shand, and Olivier Bonaventure. “Disruption Free Topology Reconfiguration in OSPF Networks”. In: *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*. IEEE. 2007, pp. 89–97.

[71] Benjamin Frank, Ingmar Poese, Georgios Smaragdakis, Steve Uhlig, and Anja Feldmann. *Content-aware Traffic Engineering*. Vol. 40. 1. ACM, 2012.

[72] Benjamin Frank, Ingmar Poese, Yin Lin, Georgios Smaragdakis, Anja Feldmann, Bruce Maggs, Jannis Rake, Steve Uhlig, and Rick Weber. “Pushing CDN-ISP Collaboration to the Limit”. In: *ACM SIGCOMM CCR* 43.3 (2013), pp. 34–44.

[73] Chris GauthierDickey and Christian Grothoff. “Bootstrapping of peer-to-peer networks”. In: *Applications and the Internet, 2008. SAINT 2008. International Symposium on*. IEEE. 2008, pp. 205–208.

[74] Adam Shaked Gish, Yuval Shavitt, and Tomer Tankel. “Geographical Statistics and Characteristics of P2P Query Strings.” In: *IPTPS*. 2007.

[75] BENOCS GmbH. *BENOCS Analytics*. 2018. URL: https://www.benocs.com/analytics/.

[76] BENOCS GmbH. *BENOCS Director*. 2018. URL: https://www.benocs.com/director/.

[77] yWorks GmbH. *yWorks*. URL: https://www.yworks.com/.

[78] GNU.org. *GNU cflow*. 2002. URL: https://www.gnu.org/software/cflow/manual/cflow.html.

[79] Greg Goth. “ISP Traffic Management: Will Innovation or Regulation Ensure Fairness?” In: *IEEE Distributed Systems Online* 9 (2008), p. 2.

[80] Krishna P Gummadi, Richard J Dunn, Stefan Saroiu, Steven D Gribble, Henry M Levy, and John Zahorjan. “Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload”. In: *ACM SIGOPS Operating Systems Review* 37.5 (2003), pp. 314–329.

[81] Anders Gunnar, Mikael Johansson, and Thomas Telkamp. “Traffic matrix estimation on a large IP backbone: a comparison on real data”. In: *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. ACM. 2004, pp. 149–160.

[82] Bill Pringlemeir Hans deGraaff. *gtk-gnutella manual*. 2007. URL: http://gtk-gnutella.sourceforge.net/manual/.

[83] Oliver M Heckmann. *The competitive Internet service provider: network architecture, interconnection, traffic engineering and network design*. John Wiley & Sons, 2007.

[84] C. Hedrick. *Routing Information Protocol*. RFC 1058. RFC Editor, 1988. URL: http://www.rfc-editor.org/rfc/rfc1058.txt.

[85] Félix Hernández-Campos, F Donelson Smith, and Kevin Jeffay. "Generating Realistic TCP Workloads." In: *Int. CMG conference*. 2004, pp. 273–284.

[86] Kevin Ho, Jie Wu, and John Sum. "On the session lifetime distribution of Gnutella". In: *The International Journal of Parallel, Emergent and Distributed Systems* 23.1 (2008), pp. 1–15.

[87] Liu Hongbo, Ajith Abraham, and Youakim Badr. *Neighbor Selection in Peer-to-Peer Overlay Networks: A Swarm Intelligence Approach in Pervasive Computing*. 2008.

[88] M. Horowitz and S. Lunt. *FTP Security Extensions*. RFC 2228. RFC Editor, 1997.

[89] Tobias Hoßfeld, Frank Lehrieder, David Hock, Simon Oechsner, Zoran Despotovic, Wolfgang Kellerer, and Maximilian Michel. "Characterization of BitTorrent swarms and their distribution in the Internet". In: *Computer Networks* 55.5 (2011), pp. 1197–1215.

[90] Tobias Hoßfeld, David Hock, Simon Oechsner, Frank Lehrieder, Zoran Despotovic, Wolfgang Kellerer, and Maximilian Michel. "Measurement of BitTorrent swarms and their AS topologies". In: *Computer Networks* (2009).

[91] Hung-Chang Hsiao, Hao Liao, and Cheng-Chyun Huang. "Resolving the topology mismatch problem in unstructured peer-to-peer networks". In: *IEEE Transactions on Parallel and Distributed Systems* 20.11 (2009), pp. 1668–1681.

[92] Yan Huang, Tom ZJ Fu, Dah-Ming Chiu, John Lui, and Cheng Huang. "Challenges, design and analysis of a large-scale p2p-vod system". In: *ACM SIGCOMM computer communication review* 38.4 (2008), pp. 375–388.

[93] Ralf Huelsermann, Matthias Gunkel, Clara Meusburger, and Dominic A Schupke. "Cost Modeling and Evaluation of Capital Expenditures in Optical Multilayer Networks". In: *Journal of Optical Networking* 7.9 (2008), pp. 814–833.

[94] Danny Hughes, Geoff Coulson, and James Walkerdine. "Free riding on Gnutella revisited: the bell tolls?" In: *IEEE distributed systems online* 6.6 (2005).

[95] Ralf Hulsermann, A Betker, M Jager, S Bodamer, M Barry, J Spath, Ch Gauger, and M Kohn. "A set of typical transport network scenarios for network modelling". In: *ITG FACHBERICHT* 182 (2004), pp. 65–72.

[96] Geoff Huston. *BGP Statistics from Route-Views Data*. URL: http://bgp.potaroo.net/bgprpts/rva-index.html.

[97] Gianluca Iannaccone, Chen-nee Chuah, Richard Mortier, Supratik Bhattacharyya, and Christophe Diot. "Analysis of link failures in an IP backbone". In: *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. ACM. 2002, pp. 237–242.

[98] Cisco Systems Inc. *Building Accurate Traffic Matrices with Demand Deduction*. 2013. URL: https://community.cisco.com/t5/service-providers-documents/building-accurate-traffic-matrices-with-demand-deduction-white/ta-p/3634516?attachment-id=136736.

[99] Cisco Systems Inc. *Cisco Visual Networking Index: Forecast and Methodology, 2007-2012*. 2008.

[100] Cisco Systems Inc. *Cisco Visual Networking Index: Forecast and Methodology, 2008-2013*. 2009.

[101] Cisco Systems Inc. *Cisco Visual Networking Index: Forecast and Methodology, 2009-2014*. 2010.

[102] Cisco Systems Inc. *Cisco Visual Networking Index: Forecast and Methodology, 2010-2015*. 2011.

[103] Cisco Systems Inc. *Cisco Visual Networking Index: Forecast and Methodology, 2011-2016*. 2012.

[104] Cisco Systems Inc. *Cisco Visual Networking Index: Forecast and Methodology, 2012-2017*. 2013.

[105] Cisco Systems Inc. *Cisco Visual Networking Index: Forecast and Methodology, 2013-2018*. 2014.

[106] Cisco Systems Inc. *Cisco Visual Networking Index: Forecast and Methodology, 2014-2019*. 2015.

[107] Cisco Systems Inc. *Cisco Visual Networking Index: Forecast and Methodology, 2015-2020*. 2016.

[108] Cisco Systems Inc. *Cisco Visual Networking Index: Forecast and Methodology, 2016–2021*. URL: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf.

[109] Cisco Systems Inc. *The Zettabyte Era: Trends and Analysis*. URL: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html.

[110] Gartner Inc. *Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016*. URL: https://www.gartner.com/newsroom/id/3598917.

[111] Opnet Technologies inc. *OPNET Modeler*. 2013.

[112] GSMA Intelligence. *GSMA's real-time tracker*. URL: https://www.gsmaintelligence.com/.

[113] Institut für Internet-Sicherheit. *Internet Deutschalnd*. retrieved in 2007. URL: http://www.internet-sicherheit.de/internet-deutschland.html.

[114] ipoque. *ipoque Internet Study 2008/2009 Finds Web and Streaming Outgrows P2P Traffic*. URL: https://www.ipoque.com/media/155.

[115] T ITU. "Recommendation G.114, One-way transmission time". In: *Series G: Transmission Systems and Media, Digital Systems and Networks, Telecommunication Standardization Sector of ITU* (2000).

[116] Raj Jain and Subharthi Paul. "Network virtualization and software defined networking for cloud computing: a survey". In: *IEEE Communications Magazine* 51.11 (2013), pp. 24–31.

[117] David S. Tatel (Chief Judge). *Comcast vs FCC & US*. 2010. URL: https://www.eff.org/files/comcast_v_fcc_dc_cir_2010.pdf.

[118] B. Kantor. *BSD Rlogin*. RFC 1282. RFC Editor, 1991.

[119] Wolfgang Kellerer, Gerald Kunzmann, Rüdiger Schollmeier, and Stefan Zöls. "Structured peer-to-peer systems for telecommunications and mobile environments". In: *AEU-International Journal of Electronics and Communications* 60.1 (2006), pp. 25–29.

[120] G. Kessler and S. Shepard. *A Primer On Internet and TCP/IP Tools and Utilities*. FYI 30. RFC Editor, 1997. URL: http://www.rfc-editor.org/rfc/rfc2151.txt.

[121] Alexander Klemm, Christoph Lindemann, Mary K Vernon, and Oliver P Waldhorst. "Characterizing the query behavior in peer-to-peer file sharing systems". In: *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. ACM. 2004, pp. 55–67.

[122] J. Klensin. *Simple Mail Transfer Protocol*. RFC 5321. RFC Editor, 2008. URL: http://www.rfc-editor.org/rfc/rfc5321.txt.

[123] Tor Klingberg and Raphael Manfredi. "The gnutella protocol specification v0. 6". In: *Technical specification of the Protocol* (2002).

[124] Simon Knight, Hung X Nguyen, Nick Falkner, Rhys Bowden, and Matthew Roughan. "The internet topology zoo". In: *IEEE Journal on Selected Areas in Communications* 29.9 (2011), pp. 1765–1775.

[125] Mirko Knoll, Arno Wacker, Gregor Schiele, and Torben Weis. "Decentralized bootstrapping in pervasive applications". In: *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on*. IEEE. 2007, pp. 589–592.

[126] Eddie Kohler, Mark Handley, and Sally Floyd. *Datagram Congestion Control Protocol (DCCP)*. Tech. rep. 2006.

[127] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. "Software-defined networking: A comprehensive survey". In: *Proceedings of the IEEE* 103.1 (2015), pp. 14–76.

[128] Ramayya Krishnan, Michael D Smith, Zhulei Tang, and Rahul Telang. "The impact of free-riding on peer-to-peer networks". In: *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*. IEEE. 2004, 10–pp.

[129] James Kurose and Keith Ross. *Computer Networking: A Top-Down Approach (7th Edition)*. Pearson, 2016. ISBN: 0133594149.

[130] Craig Labovitz, Scott Iekel-Johnson, Danny McPherson, Jon Oberheide, and Farnam Jahanian. "Internet inter-domain traffic". In: *ACM SIGCOMM Computer Communication Review*. Vol. 40. 4. ACM. 2010, pp. 75–86.

[131] Lun Li, David Alderson, Walter Willinger, and John Doyle. "A first-principles approach to understanding the internet's router-level topology". In: *ACM SIGCOMM Computer Communication Review*. Vol. 34. 4. ACM. 2004, pp. 3–14.

[132] Michael Liljenstam, Jason Liu, and David M Nicol. "Simulation of large scale networks II: development of an internet backbone topology for large-scale network simulations". In: *Proceedings of the 35th conference on Winter simulation: driving innovation*. Winter Simulation Conference. 2003, pp. 694–702.

[133] Yunhao Liu, Zhenyun Zhuang, Li Xiao, and Lionel M Ni. "AOTO: adaptive overlay topology optimization in unstructured P2P systems". In: *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*. Vol. 7. IEEE. 2003, pp. 4186–4190.

[134] Yunhao Liu, Li Xiao, A-H Esfahanian, and Lionel M Ni. "Approaching optimal peer-to-peer overlays". In: *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2005. 13th IEEE International Symposium on*. IEEE. 2005, pp. 407–414.

[135] Virginia Lo, Dayi Zhou, Yuhong Liu, Chris GauthierDickey, and Jun Li. "Scalable supernode selection in peer-to-peer overlay networks". In: *Hot Topics in Peer-to-Peer Systems, 2005. HOT-P2P 2005. Second International Workshop on*. IEEE. 2005, pp. 18–25.

[136] Richard Macmanus. *Trend Watch: P2P Traffic Much Bigger Than Web Traffic)*. URL: https://readwrite.com/2006/12/06/p2p_growth_trend_watch/.

[137] Gregor Maier, Anja Feldmann, Vern Paxson, and Mark Allman. "On dominant characteristics of residential broadband internet traffic". In: *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*. ACM. 2009, pp. 90–102.

[138] Andrew Makhorin. "GLPK (GNU linear programming kit)". In: (2017). URL: http://www.gnu.org/s/glpk/glpk.html.

[139] Sergio Marti and Hector Garcia-Molina. "Taxonomy of trust: Categorizing P2P reputation systems". In: *Computer Networks* 50.4 (2006), pp. 472–484.

[140] Nick McKeown. "Software-defined networking". In: *INFOCOM keynote talk* 17.2 (2009), pp. 30–32.

[141] Bernhard Meindl and Matthias Templ. "Analysis of commercial and free and open source solvers for linear optimization problems". In: *Eurostat and Statistics Netherlands within the project ESSnet on common tools and harmonised methodology for SDC in the ESS* 20 (2012).

[142] Hans Mittelmann. "Benchmarks for Optimization Software". In: (2017). URL: http://plato.asu.edu/bench.html.

[143] P. Mockapetris. *Domain names - implementation and specification*. STD 13. RFC Editor, 1987. URL: http://www.rfc-editor.org/rfc/rfc1035.txt.

[144] A. Morton, G. Ramachandran, and G. Maguluri. *Reporting IP Network Performance Metrics: Different Points of View*. RFC 6703. RFC Editor, 2012. URL: http://www.rfc-editor.org/rfc/rfc6703.txt.

[145] John Moy. *OSPF version 2*. Tech. rep. 1997.

[146] Wolfgang Muehlbauer, Anja Feldmann, Olaf Maennel, Matthew Roughan, and Steve Uhlig. "Building an AS-topology model that captures route diversity". In: *ACM SIGCOMM Computer Communication Review* 36.4 (2006), pp. 195–206.

[147] Katta G Murty. *Linear programming*. Vol. 60. Wiley New York, 1983.

[148] Juniper Networks. *100-Gigabit DWDM OTN PIC Integrated Transceiver Optical Interface Specifications*. URL: https://www.juniper.net/documentation/en_US/release-independent/junos/topics/reference/specifications/pic-ptx-series-dwdm-otn-100-gbps-optical-specifications.html.

[149] Juniper Networks. *Juniper Flow Monitoring*. 2011. URL: https://www.juniper.net/us/en/local/pdf/app-notes/3500204-en.pdf.

[150] Juniper Networks. *Juniper PTX Series Packet Transport Routers*. URL: https://www.juniper.net/us/en/products-services/routing/ptx-series/.

[151] Juniper Networks. *Juniper T4000 Core Router*. URL: https://www.juniper.net/documentation/en_US/release-independent/junos/information-products/pathway-pages/t-series/t4000/index.html.

[152] Antonio Nucci and Konstantina Papagiannaki. *Design, measurement and management of large-scale IP networks: Bridging the gap between theory and practice*. Cambridge University Press, 2009.

[153] Bruno Astuto A Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti. "A survey of software-defined networking: Past, present, and future of programmable networks". In: *IEEE Communications Surveys & Tutorials* 16.3 (2014), pp. 1617–1634.

[154] Gurobi Optimization. *Gurobi Optimizer version 7.0. 2*. 2017.

[155] Gurobi Optimzation. *Gurobi Optimization - The State-of-the-Art Mathematical Programming Solver*. URL: https://www.gurobi.com/.

[156] David Oran. *OSI IS-IS intra-domain routing protocol*. Tech. rep. 1990.

[157] Subharthi Paul, Jianli Pan, and Raj Jain. "Architectures for the future networks and the next generation Internet: A survey". In: *Computer Communications* 34.1 (2011), pp. 2–42.

[158] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. *Framework for IP Performance Metrics*. RFC 2330. RFC Editor, 1998. URL: http://www.rfc-editor.org/rfc/rfc2330.txt.

[159] Peter Phaal, Sonia Panchen, and Neil McKee. *RFC 3176: sFlow*. 2001.

[160] Ingmar Poese, Benjamin Frank, Georgios Smaragdakis, Steve Uhlig, Anja Feldmann, and Bruce Maggs. "Enabling Content-aware Traffic Engineering". In: *ACM SIGCOMM Computer Communication Review* 42.5 (2012), pp. 21–28.

[161] Ingmar Poese, Benjamin Frank, Bernhard Ager, Georgios Smaragdakis, and Anja Feldmann. "Improving Content Delivery using Provider-aided Distance Information". In: *Proceedings of ACM IMC 2010*. Melbourne, Australia, 2010.

[162] Ingmar Poese, Benjamin Frank, Bernhard Ager, Georgios Smaragdakis, Steve Uhlig, and Anja Feldmann. "Improving Content Delivery with PaDIS". In: *IEEE Internet Computing* (2012). ISSN: 1089-7801. DOI: http://dx.doi.org/10.1109/MIC.2011.105.

[163] J. Postel. *Internet Control Message Protocol*. STD 5. RFC Editor, 1981. URL: http://www.rfc-editor.org/rfc/rfc792.txt.

[164] J. Postel and J. Reynolds. *File Transfer Protocol*. STD 9. RFC Editor, 1985. URL: http://www.rfc-editor.org/rfc/rfc959.txt.

[165] J. Postel and J. Reynolds. *Telnet Protocol Specification*. STD 8. RFC Editor, 1983. URL: http://www.rfc-editor.org/rfc/rfc854.txt.

[166] Aiko Pras, Lambert Nieuwenhuis, Remco van de Meent, and Michel Mandjes. "Dimensioning network links: A new look at equivalent bandwidth". In: *IEEE network* 23.2 (2009), pp. 5–10.

[167] Bruno Quoitin, Steve Uhlig, and Olivier Bonaventure. "Using redistribution communities for interdomain traffic engineering". In: *From QoS Provisioning to QoS Charging*. Springer, 2002, pp. 125–134.

[168] Lakshmish Ramaswamy and Ling Liu. "Free riding: A new challenge to peer-to-peer file sharing systems". In: *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*. IEEE. 2003, 10–pp.

[169] Yakov Rekhter, Tony Li, and Susan Hares. *A border gateway protocol 4 (BGP-4)*. Tech. rep. 2005.

[170] E. Rescorla. *HTTP Over TLS*. RFC 2818. RFC Editor, 2000. URL: http://www.rfc-editor.org/rfc/rfc2818.txt.

[171] E. Rescorla and A. Schiffman. *The Secure HyperText Transfer Protocol*. RFC 2660. RFC Editor, 1999.

[172] Sean Rhea, Brighten Godfrey, Brad Karp, John Kubiatowicz, Sylvia Ratnasamy, Scott Shenker, Ion Stoica, and Harlan Yu. "OpenDHT: a public DHT service and its uses". In: *ACM SIGCOMM Computer Communication Review*. Vol. 35. 4. ACM. 2005, pp. 73–84.

[173] R. T. Rockafellar. *Fundamntals Of Optimization, Lecture Notes 2007*. URL: https://sites.math.washington.edu/~rtr/fundamentals.pdf.

[174] Timothy Roscoe. "The end of Internet architecture". In: *Proceedings of the 5th Workshop on Hot Topics in Networks*. Irvine, CA, USA. 2006.

[175] E. Rosen, A. Viswanathan, and R. Callon. *Multiprotocol Label Switching Architecture*. RFC 3031. RFC Editor, 2001. URL: http://www.rfc-editor.org/rfc/rfc3031.txt.

[176] Stefan Saroiu, P Krishna Gummadi, and Steven D Gribble. "Measurement study of peer-to-peer file sharing systems". In: *Multimedia Computing and Networking 2002*. Vol. 4673. International Society for Optics and Photonics. 2001, pp. 156–171.

[177] Donnie Savage, Donald Slice, Russ White, James Ng, Peter Paluch, and Steven Moore. "Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP)". In: (2016).

[178] Adarshpal S Sethi and Vasil Y Hnatyshin. *The practical OPNET user guide for computer network simulation*. Chapman and Hall/CRC, 2012.

[179] sFlow.org. *Traffic Monitoring using sFlow*. 2003. URL: https://sflow.org/sFlowOverview.pdf.

[180] Georgios Smaragdakis. "Overlay Network Creation and Maintenance with Selfish Users". Ph.D. Dissertation. Boston, MA: Boston University, Computer Science Department, 2008. URL: http://www.smaragdakis.net/publications/Smaragdakis-Dissertation-2008/Smaragdakis-Dissertation-2008.pdf.

[181] K. Sollins. *The TFTP Protocol (Revision 2)*. STD 33. RFC Editor, 1992.

[182] Augustin Soule, Anukool Lakhina, Nina Taft, Konstantina Papagiannaki, Kave Salamatian, Antonio Nucci, Mark Crovella, and Christophe Diot. "Traffic matrices: balancing measurements, inference and modeling". In: *ACM SIGMETRICS Performance Evaluation Review*. Vol. 33. 1. ACM. 2005, pp. 362–373.

[183] Neil Spring, Ratul Mahajan, and David Wetherall. "Measuring ISP topologies with Rocketfuel". In: *ACM SIGCOMM Computer Communication Review* 32.4 (2002), pp. 133–145.

[184] Dimitri Staessens, Didier Colle, Ilse Lievens, Mario Pickavet, and Piet Demeester. "Path Protection in Transparent Networks". In: *Proceedings of the 1st KEIO and Gent University G-COE Joint workshop for future network 2008*. 2008, pp. 17–20.

[185] William Stallings. *Data and Computer Communications*. Vol. 10. Pearson, 2013.

[186] Statista. *Global digital population as of August 2017 (in millions)*. URL: https://www.statista.com/statistics/617136/digital-population-worldwide/.

[187] Internet Live Stats. *Internet Users*. URL: http://www.internetlivestats.com/internet-users/.

[188] Martin Stiemerling, S Kiesel, M Scharf, H Seidel, and S Previdi. *Application-Layer Traffic Optimization (ALTO) Deployment Considerations*. Tech. rep. 2016.

[189] Daniel Stutzbach and Reza Rejaie. "Characterizing today's Gnutella topology". In: *Technical Report CIS-TR-04–02* (2004).

[190] Daniel Stutzbach and Reza Rejaie. "Understanding churn in peer-to-peer networks". In: *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. ACM. 2006, pp. 189–202.

[191] Cisco Systems. *An Introduction to IGRP*. URL: https://www.cisco.com/c/en/us/support/docs/ip/interior-gateway-routing-protocol-igrp/26825-5.html.

[192] Cisco Systems. *Best Practices in Core Network Capacity Planning*. URL: https://www.cisco.com/c/en/us/products/collateral/routers/wan-automation-engine/white_paper_c11-728551.pdf.

[193] Cisco Systems. *Cisco Carrier Routing System*. URL: https://www.cisco.com/c/en/us/products/routers/carrier-routing-system/index.html.

[194] Cisco Systems. *Cisco CPAK 100GBASE Modules Data Sheet*. URL: https://www.cisco.com/c/en/us/products/collateral/interfaces-modules/transceiver-modules/data_sheet_c78-728110.html.

[195] Cisco Systems. *Cisco nLight^{TM} Technology: A Multi-Layer Control Plane Architecture for IP and Optical Convergence*. 2012. URL: https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-3750-series-switches/whitepaper_c11-718852.html.

[196] Cisco Systems. *Converged Transport Architecture: Improving Scale and Efficiency in Service Provider Backbone Networks*. URL: https://www.cisco.com/c/en/us/products/collateral/routers/carrier-routing-system/white_paper_c11-728242.html.

[197] Cisco Systems. *New Cisco CRS Elastic Core Solution Improves Internet Efficiency with Programmability*. 2012. URL: https://newsroom.cisco.com/press-release-content?type=webcontent&articleId=1042271.

[198] Riverbed Systems. *OPNET Technologies - Network Simulator | Riverbed*. URL: https://www.riverbed.com/sg/products/steelcentral/opnet.html.

[199] Tim Szigeti, Christina Hattingh, Robert Barton, and Kenneth Briley Jr. *End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks*. Cisco Press, 2013.

[200] Techopedia. *Secure Copy*. URL: https://www.techopedia.com/definition/26142/secure-copy.

[201] Thomas Telkamp. "Traffic characteristics and network planning". In: *NANOG 2002* (2002).

[202] D. Thaler and C. Hopps. *Multipath Issues in Unicast and Multicast Next-Hop Selection*. RFC 2991. RFC Editor, 2000. URL: http://www.rfc-editor.org/rfc/rfc2991.txt.

[203] Jing Tian and Yafei Dai. "Understanding the Dynamic of P2P Systems". In: *International Workshop on P2P Systems (IPTPS)*. 2007.

[204] Michael Tuexen and Randall R Stewart. "Stream Control Transmission Protocol (SCTP) Chunk Flags Registration". In: *RFC 4960* (2011).

[205] Paul Tune and Matthew Roughan. "Internet traffic matrices: A primer". In: *Advances in Networking, Vol. 1. ACM SIGCOMM* (2013).

[206] Steve Uhlig. "On the complexity of Internet traffic dynamics on its topology". In: *Telecommunication Systems* 43.3-4 (2010), pp. 167–180.

[207] Sandvine Incorporated ULC. *2016 Global Internet Phenomena Report - Latin America and North America*. URL: https://www.sandvine.com/downloads/general/global-internet-phenomena/2016/global-internet-phenomena-report-latin-america-and-north-america.pdf.

[208] Sandvine Incorporated ULC. *Global Internet Phenomena Spotlight - Netflix Rising*. 2011.

[209] European Union. *General Data Protection Regulation*. 2016. URL: https://gdpr-info.eu/.

[210] S. Waldbusser. *Remote Network Monitoring Management Information Base*. STD 59. RFC Editor, 2000. URL: http://www.rfc-editor.org/rfc/rfc2819.txt.

[211] S. Waldbusser. *Remote Network Monitoring Management Information Base Version 2 using SMIv2*. RFC 2021. RFC Editor, 1997. URL: http://www.rfc-editor.org/rfc/rfc2021.txt.

[212] Klaus Wehrle, Mesut Günes, and James Gross. *Modeling and tools for network simulation*. Springer Science & Business Media, 2010.

[213] Laurence Wolsey. "Mixed Integer Programming". In: *Wiley Encyclopedia of Computer Science and Engineering* (2008).

[214] Haiyong Xie, Y Richard Yang, Arvind Krishnamurthy, Yanbin Grace Liu, and Abraham Silberschatz. "P4P: Provider Portal for (P2P) Applications". In: *ACM SIGCOMM Computer Communication Review* 38.4 (2008), pp. 351–362.

[215] T. Ylonen and C. Lonvick. *The Secure Shell (SSH) Transport Layer Protocol*. RFC 4253. http://www.rfc-editor.org/rfc/rfc4253.txt. RFC Editor, 2006. URL: http://www.rfc-editor.org/rfc/rfc4253.txt.

[216] Hongliang Yu, Dongdong Zheng, Ben Y Zhao, and Weimin Zheng. "Understanding user behavior in large-scale video-on-demand systems". In: *ACM SIGOPS Operating Systems Review*. Vol. 40. 4. ACM. 2006, pp. 333–344.

[217] zdnet.com. *NBN considers throttling 'extreme' fixed-wireless users*. 2018. URL: https://www.zdnet.com/article/nbn-considers-throttling-extreme-fixed-wireless-users/.

[218] Manaf Zghaibeh and Kostas G Anagnostakis. "On the impact of p2p incentive mechanisms on user behavior". In: *NetEcon+ IBC* (2007).

[219] Beichuan Zhang, Raymond Liu, Daniel Massey, and Lixia Zhang. "Collecting the Internet AS-level topology". In: *ACM SIGCOMM Computer Communication Review* 35.1 (2005), pp. 53–61.

[220] Rui Zhang-Shen and Nick McKeown. "Designing a predictable internet backbone with valiant load-balancing". In: *International Workshop on Quality of Service*. Springer. 2005, pp. 178–192.

[221] Han Zheng, Eng Keong Lua, Marcelo Pias, and Timothy G Griffin. "Internet routing policies and round-trip-times". In: *International Workshop on Passive and Active Network Measurement*. Springer. 2005, pp. 236–250.

[222] Thomas Zinner, Stefan Geissler, Stanislav Lange, Steffen Gebert, Michael Seufert, and Phuoc Tran-Gia. "A discrete-time model for optimizing the processing time of virtualized network functions". In: *Computer Networks* 125 (2017), pp. 4–14.