Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Telecommunication Engineering

On Bandwidth-Delay-Constrained Least-Delay-Variation Problem in Smart Grid Ethernet Networks

Doctoral thesis

Ing. Tomáš Hégr

Ph.D. programme: P2612 Electrical Engineering and Information Technology
Branch of study: 2601V013 Telecommunication Engineering
Supervisor: Doc. Ing. Leoš Boháč, Ph.D.

Prague, August 2019

Thesis Supervisor:
Doc. Ing. Leoš Boháč, Ph.D.
Department of Telecommunication Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2
160 00 Prague 6
Czech Republic

Declaration

I hereby declare I have written this doctoral thesis independently and quoted all the sources of information used in accordance with methodological instructions on ethical principles for writing an academic thesis. Moreover, I state that this thesis has neither been submitted nor accepted for any other degree.

In Prague, August 2019

.....
Ing. Tomáš Hégr

Abstract

The main goal of the doctoral thesis is to design and verify an optimization algorithm for constrained multicast communication in a local data network run in power substations, as defined by IEC 61850. The Software-Defined Networking (SDN) concept provides the opportunity to fully utilize redundant network topologies for multicast communication, typically limited only to the minimum spanning tree.

The optimization task addresses the potential of advanced control over multiple multicast groups sharing an underlying infrastructure constrained by available bandwidth and maximum transfer time with the objective to minimize delay variation among multicast subscribers. **The task is abstracted into a multi-tree Bandwidth-Delay-Constrained Least-Delay-Variation (BDLDV) problem, a sort of Steiner tree problem in networks which is proven to be NP-complete.**

A decomposed linear program, initialized by solutions obtained from a Genetic Algorithm (GA), is proposed to tackle the problem's scalability. The analysis shows that this hybrid approach can deliver optimal forwarding configurations for instances of real-world size. The proposed model improves scalability in comparison to the regular Integer Linear Programming (ILP) model preserving the high quality and stability of delivered solutions. Eventually, simulations confirm the usability of optimized multicast configurations in non-deterministic Ethernet networks.

The main contributions of the doctoral thesis are following.

- The design of a methodology for the measurement of the switch fabric latency.
- The formalization of the Least-Delay-Variation (LDV) multicast problem, and an ILP model accompanied by the structural analysis of the obtained results.
- The design and evaluation of the GA that significantly improves the scalability of the proposed ILP solution.
- The synthesis of previous results into a final decomposed model for a multi-tree BDLDV problem, including a quality and stability evaluation, and the simulation of optimized configurations.

Keywords: IEC 61850, SDN, Smard Grids, Genetic Algorithm, ILP, Simulations.

Abstrakt

Cílem předložené disertační práce je návrh a ověření optimalizačního algoritmu pro vícesměrové vysílání ve specifických lokálních datových sítích energetických stanic popsaných standardem IEC 61850. Pomocí konceptu softwarově definovaných sítí lze vícesměrovou komunikaci na spojové vrstvě distribuovat cíleněji v redundantní infrastruktuře často omezené pouze minimální kostrou.

Optimalizační úloha se zabývá možností využití pokročilého řízení vícesměrového vysílání pro více skupin na sdílené infrastruktuře s omezenou přenosovou kapacitou a maximálním časem přenosu s cílem minimalizace časového rozptylu doručení zpráv mezi jednotlivými příjemci. Úloha je abstrahována do podoby rozšířeného Steinerova problému v sítích, nazvaného BDLDV problém na více stromech, který je NP-úplný.

K řešení problému této třídy složitosti byla navržena inovativní kombinace dekomponovaného modelu pro lineární programování v kombinaci s inicializačními výsledky získanými z genetického algoritmu. Jak ukazuje vyhodnocení, lze tímto hybridním přístupem nalézt optimální směrovací konfiguraci pro instance odpovídající svojí velikostí reálným datovým sítím. Zároveň je použitelnost těchto optimalizovaných konfigurací ověřena a potvrzuje svůj přínos i na úrovni nedeterminického přenosového kanálu jakým je Ethernet.

Přínosy práce spočívají v následujících bodech.

- Návrh a ověření metodiky pro měření zpoždění přepínací struktury uvnitř přepínače pro získání informace nezbytné pro provedení simulací.
- Definice a návrh lineárního programu pro problém vícesměrové komunikace s cílem minimalizace rozptylu zpráv včetně strukturální analýzy dosažených výsledků.
- Návrh a ověření genetického algoritmu významně zlepšujícího výpočetní limity lineárního programu.
- Syntéza předchozích poznatků v uceleném modelu včetně ověření stability a kvality výsledků a evaluace optimalizovaných konfigurací na simulačních modelech redundantních datových sítí.

Acknowledgments

Every large project takes a lot of resources and is built step by step. Analogously, this doctoral thesis was not born all of a sudden. It took years to put all the pieces together from the very first idea to the very last evaluation. Each such piece needed a different kind of support to fit the final goal, but the relief was always in my reach.

I want to express my gratitude to my supervisor Doc. Ing. Leoš Boháč, Ph.D., who was always resolute and uncompromising with published texts. Then, my gratitude goes to the whole 601 crew. Namely to you Zbyněk, Ondra, Peter, Radek, Ondra, and Tomáš. All of you left some footprint in my thesis, and I appreciate it.

The exceptional place in my gratitude is reserved to you Miloš. Time shows that your continuous expert support and encouragements, but sometimes even adamant remarks were the most important fuel I needed to finish my Ph.D. studies.

I am thankful for the support given to me by all of my colleagues, from the department or abroad, who were helpful to me whenever I needed to consult or clarify my ideas and dispel doubts. The thesis would not exist without the technical support of my department, computational support provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, and financial support of grants SGS16/158/OHK3/2T/13, SGS13/200/OHK3/3T/13, TA04011571, VG20132015104.

Lastly, I would like to thank my family that they provided me with me enough time to express myself. They always believed that there is enough determination in me to finish the thesis.

List of Tables

List of Figures

List of Acronyms

ACL Access Control List. 4, 27
ACSI Abstract Communication Service Interface. 15, 16
API Application Programming Interface. 24
ARP Address Resolution Protocol. 45, 54
ASDU Application Service Data Unit. 19
ASIC Application Specific Integrated Circuit. 42
ASN.1 Abstract Syntax Notation One. 18
ATR All Terminal Reliability. 30

BCU Bay Control Unit. 20
BDLDV Bandwidth-Delay-Constrained Least-Delay-Variation. iv, v, x, 6, 7, 33, 57, 65, 84, 110–112, 114–119, 121, 122, 124–129, 131, 133, 135–139
BER Basic Encoding Rules. 18
BFS Breadth-First Search. 86
BGP Border Gateway Protocol. 24

CAM Content-Addressable Memory. 27, 44, 55
CBT Core-Based Trees. 6
CEN European Committee for Standardization. 10
CENELEC European Committee for Standardization. 11
CER Canonical Encoding Rules. 18
CG Column Generation. 116, 118, 133, 135
CIOQ Combined Input and Output Queuing. 43
CoS Class of Service. 39, 40
CPU Central Processing Unit. 42, 43, 54, 93, 108
CRC Cyclic Redundancy Check. 47, 114
CT Current Transformer. 4
cxpb Crossover probability. 95, 140–143
cxt Crossover. 140–143

DER Distributed Energy Resources. 11, 18
DFS Depth-First Search. ix, 79, 80, 88, 91
DNP3 Distributed Network Protocol. 15
DS Data Set. 19
DUT Device Under Test. 45–47, 54
DVBMT Delay and Delay Variation-Bounded Multicast Tree. 34, 36
DVBST Delay and Delay Variation-Bounded Steiner Tree. 34, 36
DVMA Delay Variation Multicast Algorithm. 34, 35
DVMRP Distance Vector Multicast Routing Protocol. 6

EI Expected Improvement. 105

ESO European Standardisation Organisations. 10, 11
ETSI European Telecommunications Standards Institute. 11

FIB Forwarding Information Base. 3, 24, 26, 42, 125

GA Genetic Algorithm. iv, ix, 8, 36, 74–81, 84–87, 90–95, 97, 98, 102–105, 107, 108, 118, 119, 121, 122, 124, 133, 135, 137–139
GARP Generic Attribute Registration Protocol. 5
GOOSE Generic Object Oriented Substation Event. viii, 4, 5, 15–20, 29

hdl Mean population Hamming distance limit. 97, 141–143
HMI Human–Machine Interface. 20
HOL Head of Line. 43
HSR High-availability Seamless Redundancy. viii, 5, 22

ICMP Internet Control Message Protocol. 44
ICT Information and Communications Technology. 10
IEC International Electrotechnical Commission. vii, 11, 12, 14, 15, 18, 21, 122, 125, 133, 134, 136
IED Intelligent Electronic Device. viii, 14–20, 29, 30, 80, 134
IEEE Institute of Electrical and Electronics Engineers. 12
IGMP Internet Group Management Protocol. 5
ILP Integer Linear Programming. iv, 8, 36, 58, 63, 65–67, 69, 74, 108, 110, 113–115, 118–122, 124, 133, 135, 137, 138
IP Internet Protocol. 5, 24
IQ Input Queuing. 43
IQR Interquartile range. 107
IS-IS Intermediate System to Intermediate System. 21

JSON JavaScript Object Notation. 54

KMB Kou, Markowsky and Berman. 34
kp Keep best individual. ix, 99, 100, 140–143

L1 Physical layer. 42, 44–46
L2 Data link layer. 2–5, 18, 21, 47, 49, 51, 54, 134, 136
L3 Network layer. 4, 21
LAN Local Area Network. 1, 28, 29, 38, 123, 124, 136
LD Logical Device. 15
LDV Least-Delay-Variation. iv, 6, 7, 34, 35, 57, 65, 67, 70–73, 77, 81–84, 97, 104, 105, 108, 110–112, 114, 115, 119, 122, 131, 133, 135–139
li Local improvement. 98–100, 140–143
LIFO Last In First Out. 42, 44
LN Logical Node. 15
LP Liner Programming. ix, 7, 58–65, 91, 93, 115, 116, 118

MAC Media Access Control. 38, 42, 45, 47, 54
MIB Management Information Base. 15
MILP Mixed-Integer Linear Programming. 58, 63
MMRP Multiple MAC Registration Protocol. 5
MMS Manufacturing Message Specification. viii, 15, 16

MOSPF Multicast Open Shortest Path First. 6, 34
MP Master Problem. 115–118
MPLS Multiprotocol Label Switching. 24
MPLS-TE Multiprotocol Label Switching - Traffic Engineering. 24
MSMT Multiple Shared Multicast Trees. 35
MSPT Multiple Shortest Path Trees. 112, 114, 117, 125–127, 133, 136, 138
MST Minimum Spanning Tree. 5, 21, 33, 35, 110, 125–128, 131–134, 136, 138
mtsp Mutation-based terminal selection probability. 140, 141, 143
MU Merging Unit. 4, 14, 15, 19, 134
mut Mutation. 142, 143
mutpb Mutation probability. 97, 140–143
mw Objective-aware mutation. 101, 140–143

nd No-Genotype Duplicates. 98, 100, 101, 140–143
NETCONF Network Configuration Protocol. 3, 24
ng Number of generations. 140–143
NGI National Grid Infrastructure. 92, 120
NIC Network Interface Card. 55
NIST National Institute of Standards and Technology. viii, 10, 11
NOS Network Operating System. 24
NTP Network Time Protocol. 41
NVP Nominal Velocity of Propagation. 38

OF OpenFlow. ix, 23, 24, 26–28, 48, 53–56
OFCS OpenFlow Capable Switch. 26
ONF Open Networking Foundation. 26
OPL Open Programming Language. 70
OQ Output Queuing. 43, 124
OS Operating System. 24

PCEP Path Computation Element Communication Protocol. 24
PCP Priority Code Point. 125
PDH Plesiochronous Digital Hierarchy. 1
PHY Physical layer. 42
PIM-DM Protocol Independent Multicast - Dense Mode. 6
PIM-SM Protocol Independent Multicast - Sparse-Mode. 6, 35
PP Pricing Problem. 116–118, 138
PRP Parallel Redundancy Protocol. viii, 5, 21, 22
ps Population size. 140–143

QAP Quality of Alternative Paths. 29
QoS Quality of Service. 2, 4, 5, 28, 32, 34, 40, 70, 71, 73

RAM Random-Access Memory. 27
RFC Request for Comments. 42
RMP Restricted Master Problem. 116–119, 133, 135, 138
RP Rendezvous Point. 35
RSTP Rapid Spanning Tree Protocol. 21

SA Substation Automation. 4, 30, 134, 137–139

SAS Substation Automation System. 3, 11, 14, 19, 20, 29, 41, 73, 125, 139
SCADA Supervisory Control And Data Acquisition. 13–15, 19, 20
SCD Substation Configuration Description. 13
SCL Substation Configuration Language. 13, 15
SCSM Specific Communication Service Mappings. 12, 16
SCU Substation Control Unit. 20
SD Software-Defined. 4, 22
SDH Synchronous Digital Hierarchy. 1
SDN Software-Defined Networking. iv, viii, 2, 4–7, 9, 13, 22–26, 33, 53, 54, 56, 70, 73, 110, 124, 125, 134, 136, 139
SDx Software-Defined Anything. 22
SFD Start Frame Delimiter. 47, 114
SFP+ Small Form-factor Pluggable. 52
SG Smart Grid. 1, 2, 7, 9–11, 13, 25, 38, 122, 134, 139
SG-CG Smart Grid Coordination Group. 9–11
SGAM Smart Grids Architecture Model. 11
SMBO Sequential Model-Based Optimization. 105
SNMP Simple Network Management Protocol. 15
SNTP Simple Network Time Protocol. 17
SOHO Small Office Home Office. 39
SP Steiner tree Problem. 4, 6, 7, 30–34, 36
SPB Shortest Path Bridging. 6, 21
SPF Shortest Path First. 21
SPOF Single Point of Failure. 30
SPT Shortest Path Tree. 30, 65–67, 70–73, 110, 112, 114, 122, 135, 137
STP Spanning Tree Protocol. 5, 6, 21, 22
SV Sampled Value. viii, 2, 4, 5, 15–20, 122, 133, 134

TC Technical Committee. 12
TCAM Ternary Content-Addressable Memory. 26, 27, 44
TCP Transmission Control Protocol. viii, 16, 25, 28
TLV Type Length Value. 18
TPE Tree Parzen Estimators. 105
TRILL Transparent Interconnection of Lots of Links. 21
ts Tournament size. 140–143
tspb Terminal selection probability. 140–143

UCA Utility Communications Architecture. 12
UCAIug UCA International Users Group. 19
UDP User Datagram Protocol. viii, 16

VLAN Virtual Local Area Network. 4, 17, 18, 20, 52
VOQ Virtual Output Queuing. 43
VPN Virtual Private Network. 24
VT Voltage Transformer. 4

WG Working Group. 12

XML eXtensible Markup Language. 13, 124
XMPP eXtensible Messaging and Presence Protocol. 24

Contents

During applause on an open stage do not bow. It is most likely for someone else.

Jára Cimrman

1 Introduction

From today's perspective, communication in power grids is full of commodity-based networking equipment forming a modern Smart Grid infrastructure. However, in not too distant past control systems in power substations and along power lines were built solely on top of binary connections, proprietary point-to-point protocols, and the Synchronous Digital Hierarchy (SDH), or even the Plesiochronous Digital Hierarchy (PDH). The operation of modern substations requires more and more automation and digitization to efficiently and securely deliver electricity to customers. While substations operated by transmission and distribution system operators still heavily employ legacy technologies, the standardization, together with the high reliability of commodity hardware, shifts communication towards ISO/IEC/IEEE 8802-3 Ethernet, further referred as the Ethernet [1].

The Ethernet has become one of the most progressive transmission technologies at the data link layer in Local Area Networks (LANs) over the last three decades. This statement is valid even for conservative industrial networking with the advent of a switched Ethernet. The applicability of the Ethernet, in mission-critical environments like Smart Grid (SG), has been immensely studied for more than a decade showing that the non-deterministic Ethernet is feasible. For the popularity, manageability, and low cost of the Ethernet in data networks, operational technologies are getting Ethernet-orientated.

Moreover, the Ethernet is referred to as a suitable solution for a real-time control systems, as suggested by works published in the field of power engineering area [2]. Nevertheless, network devices have to support many distinct protocols and technologies to run

real-time applications on the Ethernet reliably. Typically, these protocols seek to prevent loops, increase network resilience, propagate forwarding information, or to optimize traffic flows to reach a desired Quality of Service (QoS) in the network. Although these tasks are already satisfactorily solved, the complexity of such systems requires highly specialized personnel, and troubleshooting can be tricky.

Currently, many network problems and challenges are intensively investigated employing an Software-Defined Networking (SDN) concept that has the potential to fulfill SG demands and simplify the overall network management. SDN is still a subject of research, and it spreads into many forms depending on the researcher's or vendor's specialization. Regardless of the means, SDN presents a tool that allows implementing advanced forwarding algorithms increasing an automation level and introduces innovative services in data networks.

The level of automation is even more strengthened by mechanisms contained in the IEC 61850 standard as described by Molina et al. in [3]. The network control in SG, based on SDN, is promising and supported by recent publications. The crucial problem of the Ethernet-based networks is their non-deterministic nature, but critical services run on them; therefore, researchers have focused on delivering the required level of QoS to application flows. Dorsch et al. introduced and evaluated algorithms for fast rerouting of critical data flows. The SDN concept could also help to increase SG resilience, meaning both data network and power grids, as presented in [4].

With these stimulating thoughts in mind, the author of this thesis presents a novel SDN application of an advanced multicast[1] distribution at a Data link layer (L2) on the following pages. In the context of IEC 61850, the multicast is a constant data-rate stream consisting of Sampled Values (SVs) as described by IEC 61850-9-2LE [6]. Although several protocols tackle the problem of the L2 multicast forwarding in Ethernet networks, it still relies on an underlying spanning tree that can be unsatisfactory. The spanning tree approach for complex forwarding scenarios on redundant networks typically underutilizes resources. In contrast, SDN allows implementing an arbitrarily calculated multicast tree in the network giving the possibility to optimize traffic flow according to relevant requirements.

1.1 Motivation

IEC 61850 describes, among other things, communication means and systems within the substation. The standard brought two evolution and cardinal changes to the power

[1] Multicast is an uni-directional, connectionless communication between a server, a multicast source known as publisher, and a selected set of clients, multicast receivers known as subscribers [5].

engineering world. The first one is the decoupling of an abstract data model from the transmission technology. Secondly, the standard proposes to utilize the Ethernet at L2. Although networks in Substation Automation Systems (SASs)[2] are mission-critical and low-latency environments, the Ethernet as a non-deterministic technology is applicable even in such demanding areas as was shown in [8].

Since data processing from the top of the communication stack down through all layers is time consuming, data are to be published right on the L2 to satisfy the demanding delay constraints. To minimize the transmission time even more, all critical messages are distributed as the Ethernet multicast in a publisher-subscriber manner reducing network utilization. However, the absence of higher protocol layers poses a significant limitation in many ways. There are no message acknowledgments and only a limited set of control protocols available for the L2 multicast. Despite the fact that today's switched Ethernet provides a full-duplex transmission and low-latency switching, most installations in SASs sticks to a point-to-point communication model. Such caution is justifiable when it comes to the implementation of novel technologies in the power engineering sector.

While the IEC 61850 standard specifies the date model of SAS and proposes a new communication stack, it does not deal with the underlying transmission technology in depth and relies on other standards. The decoupled approach is an advantage for the sustainability of the data model, but on the other hand, it shifts the burden of responsibility to the network designer. Each switch usually has to be configured manually to guarantee the desired transmission delay, efficiently utilize network infrastructure, and to avoid congestions. It is evident that in large-scale network infrastructures this task is very complicated and susceptible to human error.

The necessity of the automatic configuration in large-scale installations is appealing. IEC 61850 covers the configuration of the power apparatus in the network, but not network devices. This drawback can be solved in the management plane using, for example, protocols such as the XML-based Network Configuration Protocol (NETCONF) [9]. However, this is only the management part of the standard, which should not be confused with the control plane. The control plane is a part of every forwarding node, and it maintains an internal Forwarding Information Base (FIB) performing forwarding decisions at the data plane. Traditional management and control protocols are very robust, but on the other hand, the decision responsibility is left on a particular forwarding node. It is challenging to make optimal and real-time forwarding decisions from the local node's perspective. This approach is limiting especially in the context of network changes, node computing performance and network convergence speed.

[2]SAS is a compound of various technologies, methods, and equipment used for the automatic operation of substations. This includes control and protection functions [7]

Using the SDN approach, the control plane at the forwarding node is simplified and shifted toward a centralized controller which concentrates all network states and statistical data. These are accessible via abstracted methods to higher functional levels and, thanks to the underlying Software-Defined (SD) framework, it allows arbitrary flow control algorithms to be implemented. One of the potential SDN use cases is the multicast distribution in the Substation Automation (SA) networks mentioned above, formally known as Steiner tree Problem (SP) in networks [10].

1.2 Problem introduction

The initial idea of the multicast distribution in the SA network has raised several significant problems to be researched. The following lines summarize the problem in context. The technological and algorithmic background can be found in Chapter 2.

In the context of IEC 61850, a Generic Object Oriented Substation Event (GOOSE) and SV messages are distributed in the network in the multicast manner and have to be delivered from the publisher to subscribers within 3 ms [11]. Both types of messages are encapsulated directly in the Ethernet frame. Such frames are tagged with the highest priority, and the multicast broadcasting is unsolicited. It should be noted that, by default, an Ethernet switch handles multicast frames in the same way as broadcasts, i.e., frames are flooded to all ports excluding that on which the frame arrived.

From the perspective of SA network utilization, SVs are predominantly interesting. Multicast frames containing SV are generated with a high sampling rate resulting in 4000, or 1600, frames per second on a 50 Hz grid generating stream consuming up to approximately 10 Mbps bandwidth per source. Although the 1Gbps Ethernet infrastructure is becoming de facto standard nowadays, the 10 Mbps stream is still a not negligible network load and, together with the trend of the process and station network aggregation [12], it can quickly lead to a violation of the delay constraint. Moreover, it turned out that while Merging Units (MUs)[3] are able to produce a nearly coherent data stream of sampled analog values on point-to-point Ethernet links [13], the consistent delivery in networks with more hops can be disrupted. Such volatility can eventually negatively bias substation protection functions.

The required QoS is commonly reached by a combination of a Virtual Local Area Network (VLAN), proper queuing, filtering and policy enforcement in the form of Access Control Lists (ACLs). Since the multicast at L2 is not usually controlled by a variety of protocols, unlike at Network layer (L3), ACLs are one of the very few ways to avoid

[3]MU is an interface unit that accepts multiple analog Current Transformer (CT), Voltage Transformer (VT) values, and binary inputs and produces multiple time-synchronized serial uni-directional multi-drop digital point-to-point outputs to provide data communication via logical interfaces [5].

broadcast flooding in the network. It is important to note that the known Internet Group Management Protocol (IGMP) is applicable only to the Internet Protocol (IP) multicast [14]. Since the power engineering sector is very prudent, the filtering plan has to be appropriately designed, which is not a trivial task as shown in [15]. Although there exists standards concerning the registration of specific attributes at multi-port bridges, or switches using the generally accepted terminology, such as Generic Attribute Registration Protocol (GARP) or its ancestor Multiple MAC Registration Protocol (MMRP) [16], [17], these bring the following limitations.

- Restricted to the topology created by Spanning Tree Protocol (STP).
- Does not reflect any QoS requirements.
- Only a single publisher per multicast group.
- Infrequently implemented in network and end devices.

All the evidence shows that managing the L2 multicast is either complicated or limited. Control complexity and ineffectiveness become apparent when we consider large-scale installations up to thousands of end devices supporting an immense number of multicast flows. The manual configuration cannot guarantee the quick topology changes necessitated by link or node failures. The topology based on Minimum Spanning Tree (MST) in redundant networks typically under-utilize available infrastructure resources. Moreover, both techniques do not guarantee optimal multicast distribution nor any load balancing within the network. Even though solutions for assuring a high-availability of the interconnecting network exist, such as Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR) [18], these need to implement an additional communication layer. Most of the limitations could be solved implementing the SDN concept.

SDN appears to be a possible framework to facilitate network control as was described above. Since SDN is still an evolving technology, the potential deployment to a mission-critical environment must be gradual and vigilant. The SDN concept is comprised of several software-based layers except for the bottommost infrastructure layer. Software usually have shorter development periods in contrast to hardware ones, and as the overall interest of the SDN community increases, drawbacks of current SDN software platforms are to be mitigated.

On the other hand, in the case of delay-constrained applications, as the GOOSE and SV distribution, an essential part of the research interest is to examine SDN hardware. Particularly, to validate a switch fabric latency in comparison to common industrial Ethernet switches. Contemporary SDN-capable switches support as a southbound protocol only OF protocol, which serves for communication between a controller and switches. The following research considers this protocol primarily.

Back to SP in networks, the efficient utilization of network resources is often considered to be the main optimization objective preceded by the requirement for the shortest delay delivery [19]. There is no known algorithm with a polynomial time complexity for SP in graphs, and hence, in particular, none for SP in networks [10].

Different variants of SPs were addressed by many authors over the years, as detailed in 2.5. Most of the proposed algorithms had one drawback in common: be feasible for real network deployments since they were designed as centralized. Thus, all information about the network has to be usually concentrated in a computational node, represented by a network device without a proper performance. In the case of the proposed SDN approach, this concentration is also an essential part of the idea, but the computation is elevated toward the upper layers. Even though the centralized SP heuristics can produce very stable results, the traditional multicast routing protocols like Multicast Open Shortest Path First (MOSPF) [20], Distance Vector Multicast Routing Protocol (DVMRP) [21], Protocol Independent Multicast - Sparse-Mode (PIM-SM) [22], Protocol Independent Multicast - Dense Mode (PIM-DM) [23] and Core-Based Trees (CBT) [24] are distributed and based on shortest path algorithms, or reverse path multicasting. The same is true even for the successor of STP, the Shortest Path Bridging (SPB) [25] protocol, which utilizes the subtrees of the default shortest path tree. None of the protocols apply cost metrics that are functions of the utilization of network resources.

The SP presented in this thesis address an objective of a delay-variation minimization in the distribution of multicast messages from a publisher to a group of subscribers. This problem is termed as an Least-Delay-Variation (LDV) multicast problem. Taking into account bandwidth, and end-to-end delay constraints, the ultimate goal is to optimize a Bandwidth-Delay-Constrained Least-Delay-Variation (BDLDV) multicast tree in the network utilizing knowledge about delays introduced by both links and nodes. As the proposed algorithms do not construct the topology, but rely on the given network structure, the optimal result of zero delay variation cannot be guaranteed. The formal definition of the problem depends on a particular context, and thus, it is separately formulated in the corresponding chapters, Chapter 4 and Chapter 6.

1.3 Aims of the doctoral thesis

Aim 1 Describe the potential application area of the optimization algorithm in the context of contemporary Smart Grid standards.

Aim 2 Design a methodology to measure switching latency of Ethernet switches operating at high data rates.

Aim 3 Formulate an LDV optimization problem for a single multicast group, and propose an optimization algorithm delivering exact solutions.

Aim 4 Design a metaheuristic for the single multicast group LDV problem and compare results with the exact algorithm proposed in Aim 3.

Aim 5 Formulate a constrained BDLDV problem for a multiple group multicast sharing common topology. Propose and evaluate an algorithm that delivers nearly-optimal solutions for real-world-sized instances.

Aim 6 Implement simulations on redundant topologies using the measured switch fabric latencies and apply the optimized multicast forwarding configurations. Verify the results and compare all algorithms under different conditions.

1.4 Thesis outline

Since the thesis deals with more research areas, these are split into several chapters containing a self-standing methodology used in the particular context. Observations obtained in one chapter are reflected in subsequent chapters. The thesis is structured as follows.

Chapter 2 – Literature Review chapter gives deeper insight into the four topics mentioned in the Introduction. At first, Section 2.1 and Section 2.2 describe the SG environment and explain the stringent requirements laid on the communication inside power substations. Next, Section 2.3 illuminates the SDN concept and contemporary protocols used to fulfill it. As the thesis deals with graph optimization algorithms, Section 2.4 outlines some of the important topological parameters mentioned later in the thesis. Lastly, Section 2.5 summarizes research done on the SP in networks.

Chapter 3 – Latency on Switched Ethernet Networks, in the first part, summarizes all components participating in the end-to-end delay. In the second part, a methodology for measuring switch fabric latency is introduced. The methodology is experimentally tested, and results are discussed.

Chapter 4 – LP and LDV Multicast Problem initially clarifies the Liner Programming (LP) and how it can be used for flow problems in networks. Next, a mathematical model related to an SP in networks is formulated containing only a single multicast tree. The model is numerically evaluated and results related to topological features are analyzed.

Chapter 5 – Genetic Algorithm for an LDV Multicast Problem shows an alternative way of how the problem can be targeted from the perspective of evolutionary algorithms. The proposed Genetic Algorithm (GA) is evaluated, parametrically optimized, and all results are compared with results obtained in 4.

Chapter 6 – Multi-tree BDLDV Multicast Problem presents the ultimate decomposed model considering all constraints. Additionally, a compact Integer Linear Programming (ILP) model and two trivial models are introduced. Models are numerically evaluated to show its stability and scalability. Further, configurations delivered by the decomposed model are extensively verified in simulations.

Chapter 7 – Conclusion summarizes all reached results and offers an outlook for future research.

If I have seen further it is by standing
on ye sholders of Giants.

Isaac Newton

2 Literature Review

The following chapter details the application area in the context of SGs. Next, it describes an SDN as a potential instrument for the realization of advanced forwarding in local networks. Then, network topologies and related structural features are briefly introduced and, at the end of the chapter, published algorithms are investigated from the perspective of multicast optimization problems. Although the topics may give the reader the impression of inhomogeneity, these are referred in the thesis to support later claims. Methods and theory are then detailed always in the context of a particular chapter.

2.1 Smart Grids

The Smart Grid is composed of many elements from a power plant and substations to utility metering, i.e., from the production of electricity to its transmission and distribution towards a customer [26]. Since every element of the Smart Grid has to be appropriately controlled and monitored, there exists an omnipresent demand. It is a reliable communication stack meeting all requirements defined by relevant standards[1].

The CEN-CENELEC-ETSI Smart Grid Coordination Group (SG-CG) takes over the definition from European Commission mandate M/490EN which describes a Smart Grid as *an electricity network that can cost-efficiently integrate the behavior and actions of all*

[1] A standard is a technical specification approved by a recognized standardization body, with which compliance is not compulsory [27].

users connected to it – generators, consumers and those that do both in order to ensure economically efficient, sustainable power system with low losses and high levels of quality and security of supply and safety [28].

Although implementations of SG differ at the technological level across the world, the set of essential domains, roles, and conceptual models remains similar. All of the functional domains leverage somehow on an Information and Communications Technology (ICT) infrastructure to reliably deliver end-to-end and almost real-time services to a customer. From the National Institute of Standards and Technology (NIST) perspective, the functional domains are divided as depicted in Figure 2.1. Under mandate M/490EN, the SG-CG defined European conceptual domain model mappable to the NIST one.

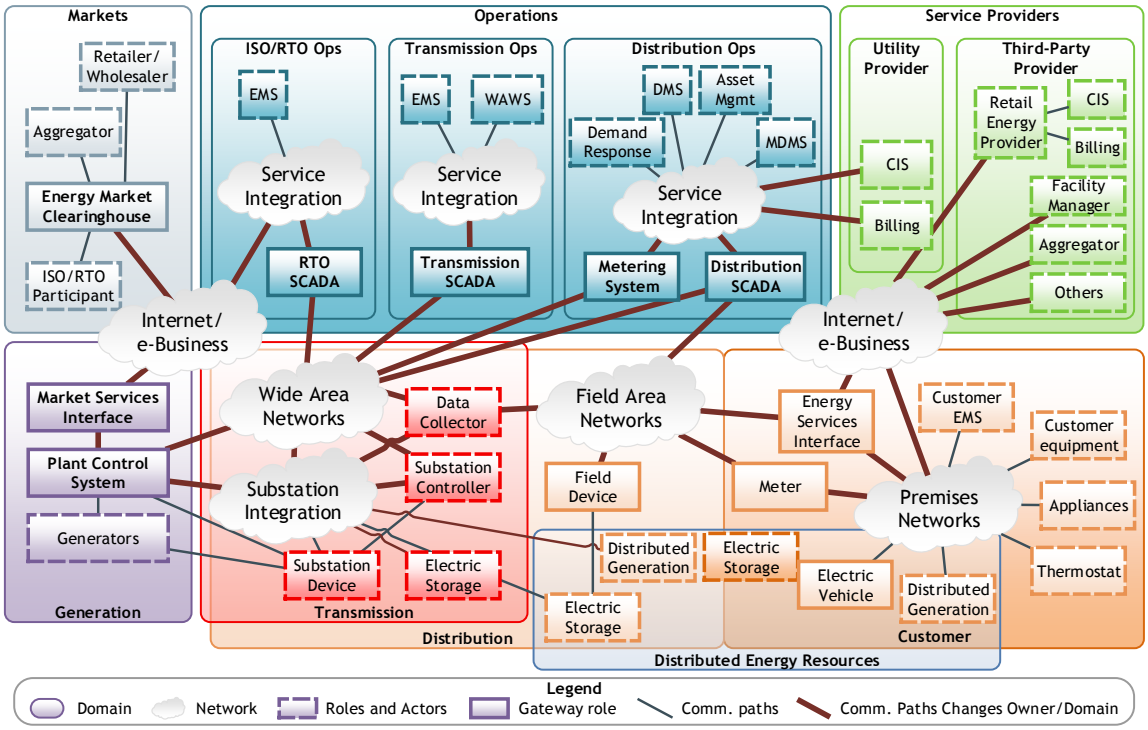


Figure 2.1: Conceptual diagram of functional domains proposed by NIST [29].

2.1.1 Standardization

Since the Smart Grid remains a hot topic in the technological and business world, many standardization bodies made use of the hype and boosted a global enthusiasm for the concept. Whereas certain bodies produce directly applicable standards, others focus on the problem integration and interoperability.

From the European perspective, the standardization effort is coordinated by the consortium of European Standardisation Organisations (ESO) - European Committee

for Standardization (CEN), European Committee for Standardization (CENELEC) and European Telecommunications Standards Institute (ETSI), forming Expert group 1 for Smart Grid standards also named SG-CG. The European Commission has issued several mandates[2] to ESO to cooperate on the development and maintenance of relevant standards. Most of the work done by SG-CG was published by the year 2014. Further work on the implementation of reference use cases in various European countries continues though.

The report [32] aims to provide a selection guide to all Smart Grid users which, depending on the targeted system and the targeted layer (component, communication or information layers), will set out the most appropriate standards to consider. In this exhaustive publication, authors proceed from the Smart Grids Architecture Model (SGAM) framework designed in [33]. The SGAM framework, depicted in Figure 2.2, and related methodology is intended to present SG use cases in technology and a solution-neutral way. As one can see in the illustration it spans over five layers representing business objectives and processes, functions, information exchange and models, communication protocols and components. Each layer covers electrical domain information management zones. The mode shows the interaction between zones and domains, but furthermore it depicts the evolution of the SG scenarios supporting the principles of universality, localization, consistency, flexibility and interoperability.

Most internationally adopted standards come from the International Electrotechnical Commission (IEC) workshop so far. Both NIST and SG-CG selected a core set of standards that almost exclusively contains IEC standards. Over 100 other documents have been identified as relevant to SG from IEC production. The very core of SG automation is IEC 61850, a backbone of the concept traversing main power grid systems.

2.2 IEC 61850 standard

The core of the SG standardization is the construction of secure and reliable smart substations which is extensively covered by the IEC 61850 standard. This standard specifies a framework, tools and methodology for the design of SASs from the perspective of system reliability and service availability relying on a decoupled data model and communication technology. The focus is put primarily on the communication and date exchange among power substations, but extensions for Distributed Energy Resources (DER), gas a and the petrochemical industry are defined as well.

[2]Mandate M/490 for smart grids [28], Mandate M/468 for electric vehicles [30], and Mandate M/441 for smart meters [31].

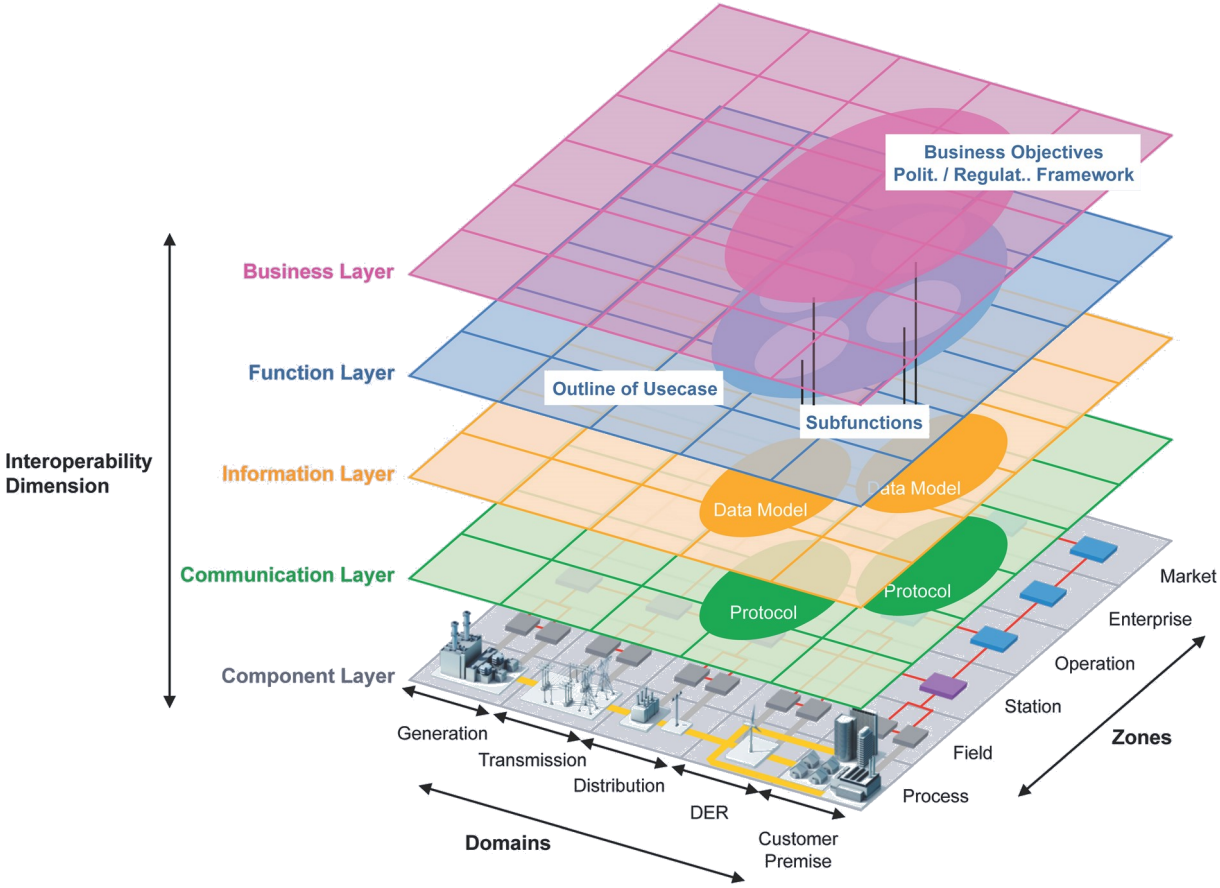


Figure 2.2: Smart Grids Architecture Model.

This standard was awaited for a long time[3] with high expectations because it unifies vendor-proprietary deployments and brings the desired interoperability. The whole standard was initially published in 10 separate parts and it was subsequently published in a second edition from 2010 to current times. Particular parts of the standard were processed and published separately on an intentional basis. Due to the complexity and international development, the standard contains some parts that are overlapping or can be misinterpreted. The second edition contributed mainly to the unification of the standard. Even though the adoption of new control technologies in power engineering is traditionally tepid, there are already large substation installations around the world implementing IEC 61850, for example, in Australia, Denmark and especially in growing China [35], [36].

[3]The standard development started in the year 1996 in Working Group (WG) 10 under Technical Committee (TC) 57 separately from the group developing Utility Communications Architecture (UCA) 2.0 in Institute of Electrical and Electronics Engineers (IEEE). In 1997, both groups agreed on cooperation. The IEC 61850 standard contains a significant portion of the UCA 2.0 specification including further extensions. These are mainly service models, object models and Specific Communication Service Mappings (SCSM) [34]

2.2.1 Tactical overview

Since the development of protocols and proprietary solutions was extremely rapid at the turn of the century, users and vendors realized the need for a unification of basic product features, tools and specifications of communication protocols. In the area of SG, WG10 was established to start the work on the standard with the following goals.

- A clear definition of communication protocols for data acquisition and the remote control of power substations. Next, a unification of data models to avoid the extensive and problematic mapping of objects in Supervisory Control And Data Acquisition (SCADA) systems.
- Long-term sustainability of any designed system on the basis of the standard thanks to the separation of communication layers and logical data model.
- An open architecture non-limiting design of the power substation, although ensuring a simplified and substantial transition from legacy systems.
- Interoperability[4] among various vendors and devices with strictly defined conformance testing.
- Interchangeability[5] to avoid the vendor lock-in effect during the whole life cycle of the power substation.

To reach these goals, the standard attempts to separate the data model from a communication model as depicted in Figure 2.3. The data model is object oriented. Application functions and interfaces were decomposed at an elementary level allowing universal elements to be defined while keeping the size of the set of elements minimal. A new Substation Configuration Language (SCL) based on eXtensible Markup Language (XML)[6] was created for substation designers and operations.

The language is an essential part of the design process, it is an instrument used to define all substation processes including the configuration of every single device and their communication tasks and demands. This approach greatly simplifies not only configuration but also automates system documentation and versioning. The SCL can be used for the description of relations in multicast trees and such a configuration can be potentially imported into an SDN controller for stationary flows. Moreover, simulation tools can draw the substation design and configuration from the Substation Configuration Description (SCD) and verify it before any real deployment.

[4] An ability of two or more devices from the same vendor, or different vendors, to exchange information and use that information for the correct execution of specified functions [5].

[5] An ability to replace a device supplied by one manufacturer with a device supplied by another manufacturer, without making changes to the other elements in the system [5].

[6] XML is a high level language that can be used to construct plain-text file formats describing application specific structured data. This enables data files to be generated and read by a computer, and which are also human legible. XML is independent of platform for example hardware, software and application, and provides free-extensibility [5].

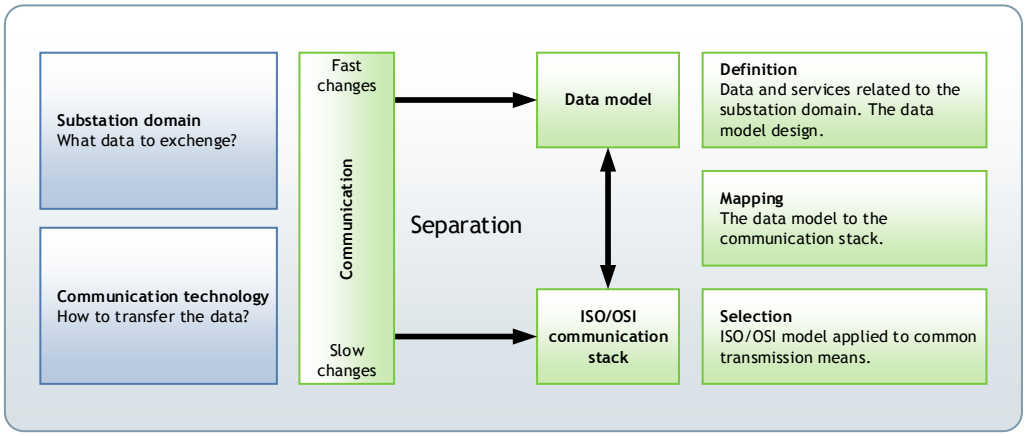


Figure 2.3: High-level schema of the decomposed communication domains to a data model and communication technology. The data model remains the same irrespective of the development of an underlaying communication technology during time.

The standard's goals are communication and control unification, long-term sustainability, free architecture, interoperability and device interchangeability avoiding the vendor lock-in problem. The most significant step in the standardization of SASs was the separation of the data model from the communication stack. This allows a data view to be independently developed for the current needs in the particular power engineering area without affecting the communication stack. The only part which is necessary to be updated is the communication stack and an appropriate mapping to payload units if the transmission technology changes. The separation and definition of the standardized data model support all benefits mentioned in the paragraph above, where the long-term sustainability of the IEC 60850 deployment is expected in tens of years.

The IEC 61850 standard is interwoven with the term Intelligent Electronic Device (IED)[7]. IEDs and MUs form an interface to primary functional equipment at a process level in a power substation. Upper functional levels of a substation (bay and station level) perform control operations on the primary equipment and provide services for data acquisition from the connected IEDs. From the standpoint of the IEC 61850 standard, the communication is not limited exclusively to an IED and SCADA, but it can be established directly between IEDs creating a room for functional automation.

Although IEDs are commonly available on the market today the transition to the new control conception based on IEC 61850, the implementation is protracted due to significant investments in legacy systems using traditional SCADA protocols (IEC 60870-5-104 or

[7]IED is any device incorporating one or more processors, with the capability to receive or send, data/control from, or to, an external source, for example electronic multi-function meters, digital relays, controllers. In the context of IEC 61850, the device capable of executing the behavior of one or more, specified logical nodes in a particular context and delimited by its interfaces [5].

Distributed Network Protocol (DNP3)). However, the IEC 61850 standard specifies more than a local intra-substation communication. It is feasible for other applications, for example data exchange between substations, control center and substations, distributed automation, exchange of metering data, log collecting, alerting and reporting.

2.2.2 Data model and services

The data model is object oriented with a tree structure similar, for example, to Management Information Base (MIB) in Simple Network Management Protocol (SNMP). Thanks to this approach it enabled the abstraction to the logical data model. The tree hierarchy offers an intuitive way to access the data. Thanks to SCL the structure is self-describing, thus, the information about services provided by an IED are accessible without any previous knowledge.

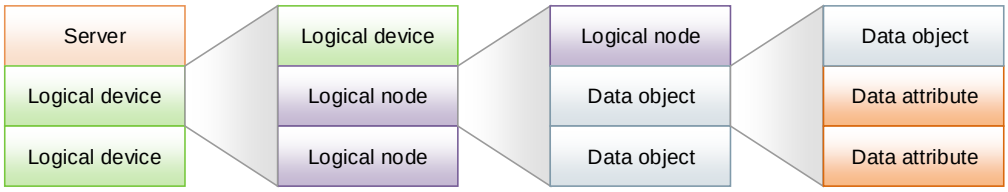


Figure 2.4: Structure of the IEC 61850 data model.

From the perspective of a client–server model, the server in IEC 61850 is always placed on the side of IED on the other side of the client is SCADA. The communication model and relation between Logical Nodes (LNs) can be extremely complex because Server can contain multiple LNs residing at more physical devices. The abstraction is a fundamental part of the whole concept. The primary standard interface is the Abstract Communication Service Interface (ACSI) serving as an access to the data objects. ACSI services mapped to Manufacturing Message Specification (MMS) enables an information exchange between a pair of Logical Devices (LDs). The available services are described in part 7-2 [37].

From the perspective of this thesis, two services are of high importance. At first, the GOOSE service, providing a fast exchange of state messages between IEDs. Secondly, the SVs, values sampled by non-conventional voltage and current transformers directly or by MU providing a digitizing interface to secondary converters. SVs and GOOSEs are an essential type of messages distributed as multicast in the network and allowing protection IEDs to take action in case of a grid malfunction.

2.2.3 Communication stack and performance classes

The communication between IEDs and other network elements is covered by a set of communication profiles built on top of an ISO/IEC 8802-3 Ethernet link layer. The communication stack was reduced in the second edition of the standard and it is shown in Figure 2.5. The unifying layer is the already mentioned SCSM that is common for all application layer protocols. The layer serves as a procedure of mapping abstract services for a particular communication profile, as detailed in parts 8. a 9. of the standard [34], [38].

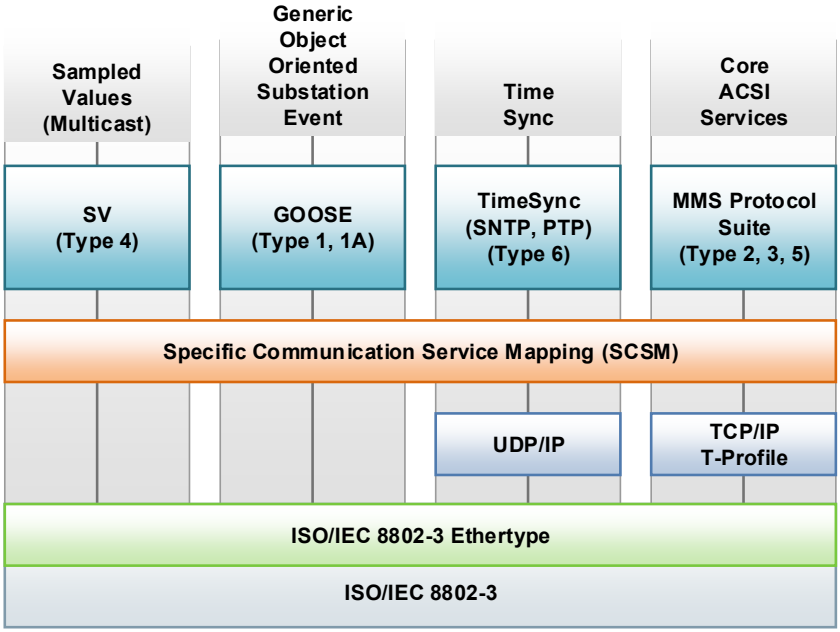


Figure 2.5: The communication stack of the second edition of IEC 61850. SV and GOOSE messages are directly mapped to the Ethernet. The optional time sync protocol is encapsulated to User Datagram Protocol (UDP). Control and report services use MMS over TCP/IP.

The standard divides available communication profiles to *A-Profile* (application profile), describing three upper layers from the ISO/OSI model, and *T-Profile* (transport profile) covering the lower four layers. The *T-Profile* is different for GOOSE and MMS. The connection oriented *T-Profile* relies on Transmission Control Protocol (TCP).

The MMS[8] protocol is used for all other types of transmissions as are configurations, logs, reports and data pulling. It does not specify operations for specific applications, which is the reason why ACSI are mapped to MMS messages. The communication is based on the client-server model, as preferred by IEC 61850.

[8]MMS came into existence in 1980 at General Motors and lately it was taken over by Boeing. In 2003, MMS was standardized as ISO/IEC 9506-1 and 2 [39], [40]. It defines a set of standardized objects, messages and encoding rules.

Two of the communication profiles GOOSE and SV are directly mapped to the link layer, i.e., as the payload of Ethernet frames. Both message types put the highest requirements on transmission times and so exploit two native Ethernet features: VLAN and priority tagging. Since these messages are a specific type of communication, they have assigned EtherType together with reserved multicast MAC address space.

As a time synchronization protocol, it was originally specified Simple Network Time Protocol (SNTP). However, IEDs are usually synchronized via either dedicated wiring as IRIG-B, 1PPS, or IEEE 1588 sharing the Ethernet substation network topology with other applications. It is expected that most of the methods will be superseded by the enhanced second version od IEEE 1588 in future installations [41], [42]. In the recent past, a new IEEE 1588 profile was published to allow compliance with the highest synchronization classes of IEC 61850 [43]. This approach guarantees device synchronization in the order of a microsecond that is necessary for applications counting on fast messages SV and GOOSE.

Each message type belonging to a given communication profile falls into a particular performance group. The performance group constrains a maximum transfer time as is shown in Figure 2.6. The transfer time is defined as the time elapsed from the moment the sending application function puts the data content on top of its transmission stack up to the moment the receiving application function extracts the data from its transmission stack.

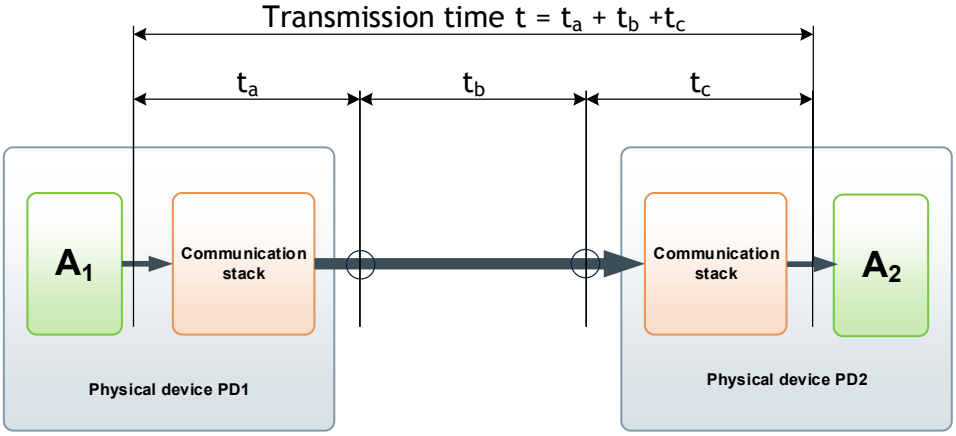


Figure 2.6: Total transmission time between two IEDs includes processing in both commutation stacks.

Time requirements are defined for seven types of messages that are listed in Table 2.1. Depending on the message type, performance classes dictate different goals; transfer time, accuracy, resolution, and sampling rate. The higher the class, the greater the requirements put on the infrastructure. The definition of performance classes and message types was slightly refined between the first and second edition of the standard. In the context of this

doctoral thesis, classes P1 and P7 are especially important with the maximum transfer time[9] limited to only 3 ms.

Table 2.1: Performance classes and message types as defined in IEC 61850 ed.2.

Message type	Maximum transfer time	Message description
1A	<3 ms, 10 ms (P1, P2)	Fast trip messages GOOSE
1B	<20 ms (P3)	Other fast messages
2	<100 ms (P4)	Medium speed messages
3	<500 ms, 1000 ms (P5, P6)	Low speed messages
4	<3 ms, 10 ms (P7∼P1, P8∼P2)	Raw data messages SV
5	<10 000 ms (P9)	File transfer functions
6	<500 ms, 1000 ms, 10 000 ms (P10∼P5, P11∼P6, P12∼P9)	Type 3 and 5 with access control

2.2.4 Critical messages

GOOSE and SV messages can be marked as critical, since they carry information essential for protection purposes. The communication is based on the publisher-subscriber model and messages are dispatched in the network as multicast traffic. Messages associate the highest priority inside a VLAN tag and the traffic is segregated into VLANs according to the assigned functional bay. The data payload is mapped directly to the Ethernet layer and is described by Abstract Syntax Notation One (ASN.1)[10] description language using Basic Encoding Rules (BER) encoding[11]. A transfer syntax of the ASN.1 language defines a particular order of data octets using given encoding rules. The message encoding is based on BER using Type Length Value (TLV) schema.

As stated in Table 2.1, both messages are limited by the highest performance requirements where the total transfer time shall be lower or equal to 3 ms [11]. It is difficult to measure the transfer time precisely as is defined, since the message timestamp includes IED's internal execution time and likely a time from the communication stack. The only part of transfer time that is possible to influece comes from the delay introduced by the message transmission along a forwarding path set up in the network, in Figure 2.6 noted as t_b. There is no acknowledgment mechanism due to the implementation at L2 and transfer time limits.

Unsolicited, unacknowledged, asynchronous GOOSE messages usually appear on both the Station and Process bus. The GOOSE transmission is connectionless. Messages are

[9]The total transmission time shall be below the order of a quarter of a cycle (5 ms for 50 Hz, 4 ms for 60 Hz) [11].

[10]ASN.1 is a flexible notation that allows one to define a variety of data types, from simple types such as integers and bit strings to structured types such as sets and sequences, as well as complex types defined in terms of others [5].

[11]BER describes how to represent or encode values of each ASN.1 type as a string of eight-bit octets [44]. The BER encoding is together with two other subsets DER a Canonical Encoding Rules (CER) defined in ITU-T X.690 [45].

transmitted in cyclic *heartbeat* periods, usually in a range from 5 to 100 ms. The receiving device can then simply decide based on the *timeAllowdToLive* parameter whether the publisher is alive or the session is broken [34]; and it can report such a fact to the superordinate SCADA system. In the case an event is invoked by the IED, a burst of messages is generated within an interval of 0.5 to 5 ms [46], to increase the probability of successful reception as depicted in Figure 2.7.

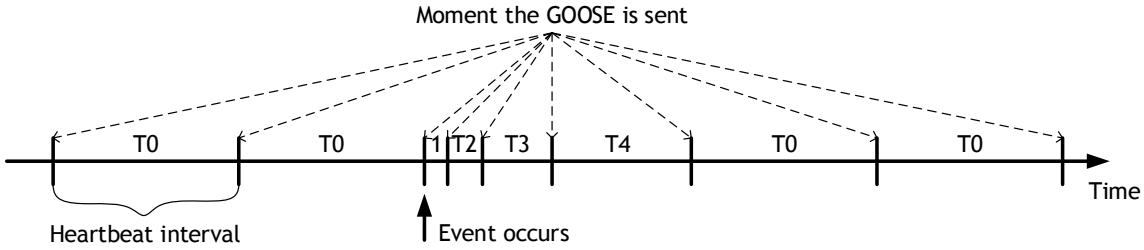


Figure 2.7: Timeline shows regular intervals (T0) in which GOOSE messages are sent between IEDs. In case an event occurs, the GOOSE burst is sent.

SV messages have similar properties as GOOSE but they are distributed strictly in the Process bus. SV messages contain raw data and are predominantly produced by MU as an unsolicited high-priority synchronous data stream. The stream profile is defined in IEC 61850-9-2LE [6] published by UCA International Users Group (UCAIug), and it is to be substituted by a backward compatible IEC 61869-9 [47].

The payload consists of Data Sets (DSs) placed in an Application Service Data Unit (ASDU), typically composed of 8 values (4 voltages and 4 current samples). Two multicast sampled value control blocks *MSVCB01* and *MSVCB02* are defined and configured. MSVCB01 has 1 ASDU and a sampling rate of 80, while *MSVCB02* has 8 ASDU and a sampling rate of 256. The sampling rates are in samples/cycle at the nominal system frequency. In the 50 Hz power system this translates to 4,000 Hz and 12,800 Hz. For example, the common stream of 4000 packets with a length of 1104 bits, a typical size including headers [13], results in 4.416 Mbps. In the second case, the stream of eight ASDUs per frame with a sampling rate of 256 can reach 9.7664 Mbps per source. Different setup can lead to even higher bandwidth requirements. Without the proper filtering multicast streams to flood the network and excessively load not only the network but also IEDs since messages have to be filtered inside the communication stack which leads to increasing the total transfer time.

2.2.5 Network and forwarding

Although the standard does not strictly advice any data network topology for SAS, it defines the structural classification of functional levels as shown in Figure 2.8. It starts

from primary functions at the Process level[12], through the Bay level[13], up to the secondary functions as operations and remote monitoring at the Station level[14].

In this sense, the classification structures the topology. The communication flows both vertically (east-west traffic) and horizontally (north-south traffic) in a substation network topology. The network is divided into two dedicated buses; the Process bus and Station bus. Usually, the division is done at the physical level, but not exclusively, since it can be done logically by VLANs to better utilize available resources. The Process bus interconnects primary equipment IEDs placed in the Process level with IEDs at the Bay level. GOOSE and SV are expected on the Process bus. Since this type of communication relies on multicast, it can cause an unwanted overload of the network in case of an improper forwarding configuration. The second Station bus interconnects the Station level with the Bay level and allows the user to access state information to the local Human–Machine Interface (HMI), remote SCADA, and other SAS. This is usually arranged by Substation Control Unit (SCU), or alternatively by the Bay Control Unit (BCU) at the lower level.

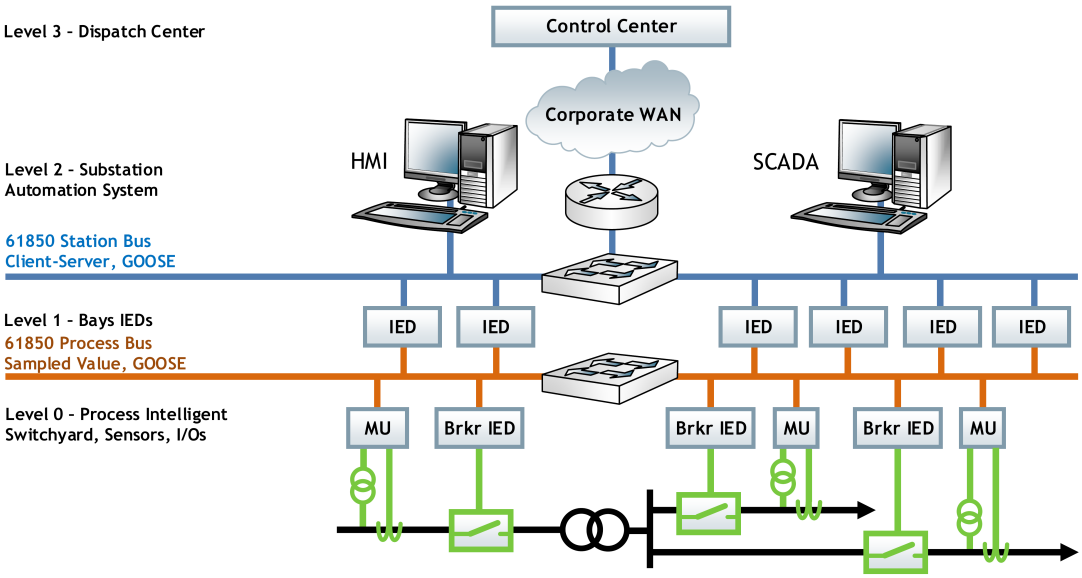


Figure 2.8: Substation architecture divided in levels according to communication requirements and related protocols.

With the increased availability requirements of data networks in mind, the topologies are designed to be highly redundant in such a demanding environment. That implies the requirement to implement a loop-prevention mechanism. Some protocols are limited only

[12]Process level functions interface to the process, i.e., binary and analogue input/output functions for example data acquisition (including sampling) and the issuing of commands. These functions communicate via logical interfaces to the bay level [5].

[13]Bay level functions use mainly the data of one bay and act mainly on the primary equipment of that bay. Bay level functions communicate within the bay level and to the process level, i.e., with any kind of remote input/output or with intelligent sensors and actuators [5].

[14]Station level functions apply to the whole substation. There are two classes of station level functions i.e., process related station level functions and interface related station level functions [5].

to specific network topologies; usually, ring, tree and meshed topologies, and some can deliver even frame loss-less operation.

Most of the protocols implemented for topology-independent networks so far are somehow based on the rock-solid STP protocol, usually due to its backward compatibility. The protocol commonly used is standardized Rapid Spanning Tree Protocol (RSTP) [16], an STP successor, providing a fast network convergence after a link failure. Based on the RSTP, proprietary protocols emerge, allowing to reconfigure the network in an order of tens of milliseconds [48].

Although the RSTP-based mechanisms for loop prevention have been widely used for years, together with the adoption of Ethernet in the power industry, new mechanisms were developed, for example, SPB[15], incorporated in IEEE 802.1q [16], and Transparent Interconnection of Lots of Links (TRILL)[16] described in RFC 7176 [50]. As the authors of [51] state, the SPB defines a set of rules to create and manage Ethernet switched networks enabling L2 multipath routing, and it has a high probability for changing how networking is also done in industrial automation. The revolutionary idea is that the multipath approach addresses problem of the underutilized, but necessarily deployed, redundant topology. Analogously, TRILL allows utilizing multiple distribution trees each one supporting the multipath forwarding [52]. The calculation of distribution trees is performed using Shortest Path First (SPF) algorithm, and although it can be adjusted by setting different link costs, it complicates the implementation of more sophisticated distribution trees. The attempt to increase the utilization of available network resources is remarkable; however, mentioned mechanisms typically follows the MST approach, when it comes to L2 multicast forwarding.

IEC a robust PRP protocol standardized in [18] that is used to reach a seamless failover mechanism in the data network. The standard proposes the truly zero-loss approach, since the PRP, a topology independent protocol, requires each end device to implement two interfaces connected to two dedicated networks. The communication goes on both networks simultaneously. This is possible because of a specially improved link layer implemented in

[15] 802.1aq Shortest Path Bridging is standardized by IEEE as an evolution of the various spanning tree protocols. 802.1aq allows for true shortest path routing, multiple equal-cost paths, much larger layer 2 topologies, faster convergence, vastly improved use of the mesh topology, single point provisioning for logical membership (E-LINE/E-LAN/E-TREE, etc.), abstraction of attached device MAC addresses from the transit devices, head end and/or transit multicast replication, all while supporting the full suite of 802.1Q [49].

[16] TRILL is an evolution of switched Ethernet networks in terms of adapting L3 routing mechanisms to L2. It is built on modified Intermediate System to Intermediate System (IS-IS) routing protocol operating on L2 that allows to construct distribution trees among special TRILL nodes knows as Routing Bridges (RBridges). These bridges encapsulate/decapsulate Ethernet frames produced/consumed by end stations to/from a TRILL header which allows to safely route the traffic via the distribution tree. The important benefit of TRILL is that it can be deployed to networks already using STP and continuously split and diminish STP-controlled parts of the network.

the device, that takes care of packet duplicates approaching both interfaces. The second option described in IEC 62439 is HSR, a ring-based protocol that should introduce lower investments as it does not require a duplicated infrastructure. In comparison to the traditional STP, it utilizes both directions in a ring topology, and thus, it delivers a virtual connection to two distinct networks from the device's perspective. Thanks to the PRP interlayer, the message duplicates from opposite directions are dropped. Devices not supporting the standard, especially in terms of the modified ISO/OSI communication stack, can be connected via a unit named Redbox that simply creates the redundant interfaces in front of the device. Ultimately, both PRP and HSR can possibly combine into one fully redundant and extremely robust solution, as is shown in Figure 2.9.

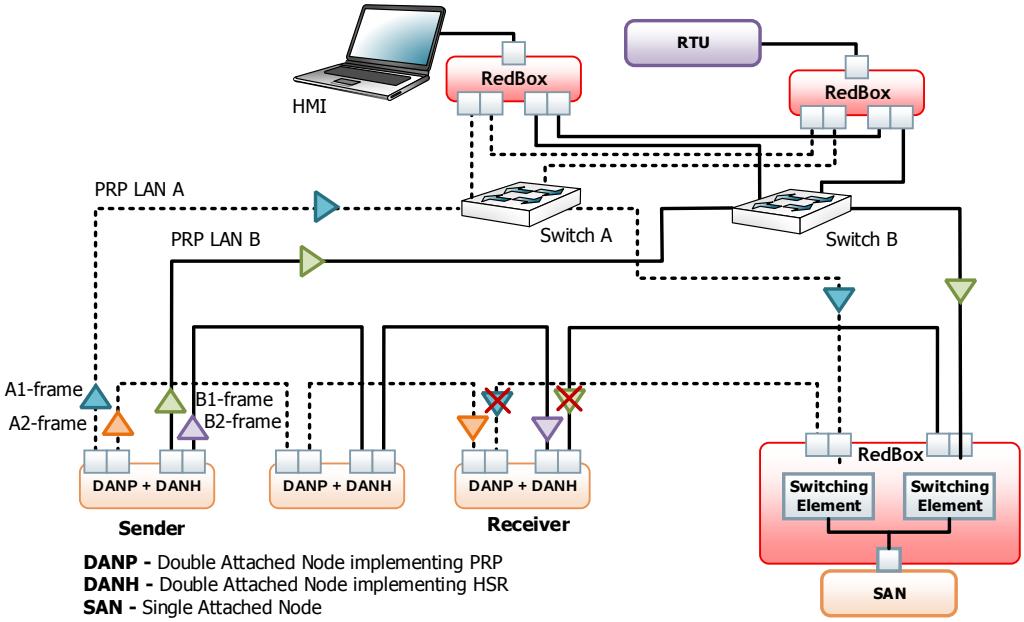


Figure 2.9: Fully redundant topology implementing an ultimate combination of PRP and HSR. The destination node consumes the first frame and drops the second.

2.3 Software-Defined Networking

As Shoshana Zuboff, Professor of the Harvard Business School, appositely stated in one of her laws: *Everything that can be automated will be automated* [53]. There are no doubts that networking stepped into the era of automation. The pressure to service-oriented networks enabling a scalable and flexible service deployment has given rise to SDN. Although SDN was one of the first SD terms that popped up in the current cloud era, it was promptly accompanied by many technologies following the very same principle, leading us to the world of Software-Defined Anything (SDx). Generally, SD can be con-

sidered as a unifying application interface separating high-level functions from lower-level heterogeneous system layers.

The SDN concept shifts the control intelligence to a logically centralized controller, thereby facilitating advanced data flow control. Moreover, the controller makes available a tremendous amount of network state data and flow statistics. Although collecting network operation data is not the primary aim of the controller, as there are many protocols for exporting monitoring data from the network, such data can be directly extracted for subsequent network optimization, analysis, diagnostics, and fine-grained flow monitoring.

Most network devices contain some kind of software delineating their behavior. The difference in the SDN concept is programmability that already existed before the hype, but the solution was usually in the category of vendor proprietary. Proprietary solutions tended to disappear in the past when networks became a critical infrastructure. That was initiated often by a lack of programming experiences, lack of solid and reliable tools, non-standardized configuration schemes implying a lack of trust in such solutions.

2.3.1 Architecture

Even though SDN is still presented as a new architecture it was established many years ago[17]. The main idea lies in the decoupling of the data and control plane, and thus enabling the abstraction of an underlying network infrastructure. The original idea was a reduction of inner complex processing parts dedicated for the control plane in a switch by moving the control responsibilities toward the controller. According to [54] and [62] the SDN concept focuses on four key features:

- Separation of the control plane from the data plane.
- Logically centralized controller providing an aggregated view of the network.
- Open interfaces between devices in the control plane (controllers) and those in the data plane.
- Applications employing open interfaces form the network logic.

A general functional architecture of SDN as is outlined in [62], and further extended in [54], is depicted in Figure 2.10. The architecture is comprised of three layers. The bottom one is named infrastructure and it is formed by switching nodes supporting one of the southbound protocols. Each node on that layer provides an access to the control plane via the southbound protocol. In reality, the way a controller instance interacts with

[17]The idea to provide out-of-the box management of forwarding in not new and it has been evolving for many years since 1996. The development documented by Sezer et al. in [54] went through the implementation of General Switch Management protocol (1996) [55], The Tempest (1998) [56], Forwarding and Control Element Separation (2004) [57], Path Computation Element (2006) [58], Ethane (2007) [59] and recently the OpenFlow (OF) protocol [60] and OnePK [61].

a network device depends on the particular implementation. While some southbound protocols allow the control plane to be accessed as for example OF, some operate in a more hybrid way[18].

The control plane is moved from the Infrastructure layer toward the upper architectural layers marked as the Controller and Application layers. The Controller layer serves as a middle layer composed of controller instances abstracting the underlying network to applications. The controller layer can be seen as Network Operating System (NOS) handling state consistency and providing a graph-like view of the network to control applications. Applications represent the intelligence of the network and they are able to upload flow rules/configurations via a northbound API down to the forwarding nodes. The flow rule is usually a set of matching rules and actions written on the node's local FIB.

In real-world implementation, the complexity is broader as the SDN disrupts the whole networking segment. SDN settles down in access, aggregation, core and data center networks and is used to control or modify forwarding behavior in both underlay, typically physical, and overlay networks.

The control centralization is very attractive since applications can heavily benefit from the collected network operational data, states and statistics. The application logic can implement advanced techniques like multi-path forwarding, path protection, congestion avoidance, etc. Moreover, the network data can be enriched by limitless meta-data sourced in the network endpoints, e.g., a network load-balancing based on an actual computation resource utilization. Although this is a significant advantage, the centralization apparently raises potential issues related primarily to the network reliability and availability.

The controller unavailability can be on the side of the switching node mitigated by running a so-called hybrid mode which ensures the same functionality as expected at an Ethernet switch. Moreover, switching nodes are to be connected to more controllers simultaneously and in the case of a connection failure they should seamlessly move to the backup connection. This feature is dependent on the southbound protocol, and it assumes

[18]Since the SDN was viewed by some vendors as a new term for eternal networking tasks, device-service provisioning systems were built simply deploying new assets according to a given template. This template-driven approach is widely used, but it does employ a management plane not the control plane. While it can be beneficial for large-scale initial deployments, it is not truly SDN as it does not provide an open unified Application Programming Interface (API). Drawbacks of such an approach are deployment unreliability, Operating System (OS) release dependencies, multi-vendor complexity and lack of transactional consistency. SDN done by configuration management addressing these challenges is for example NETCONF [9]. A certain solution situated in-between the aforementioned approaches are so called adjustment controllers. The controller here only adjusts forwarding paths using Multiprotocol Label Switching - Traffic Engineering (MPLS-TE) [63], Path Computation Element Communication Protocol (PCEP) [64] or an extended Border Gateway Protocol (BGP) [65]. Usually, this approach is limited by a destination-only forwarding model. Even a messaging protocol eXtensible Messaging and Presence Protocol (XMPP) [66] can be used to distribute commands in the network to build BGP, Multiprotocol Label Switching (MPLS), IP Virtual Private Networks (VPNs)[67].

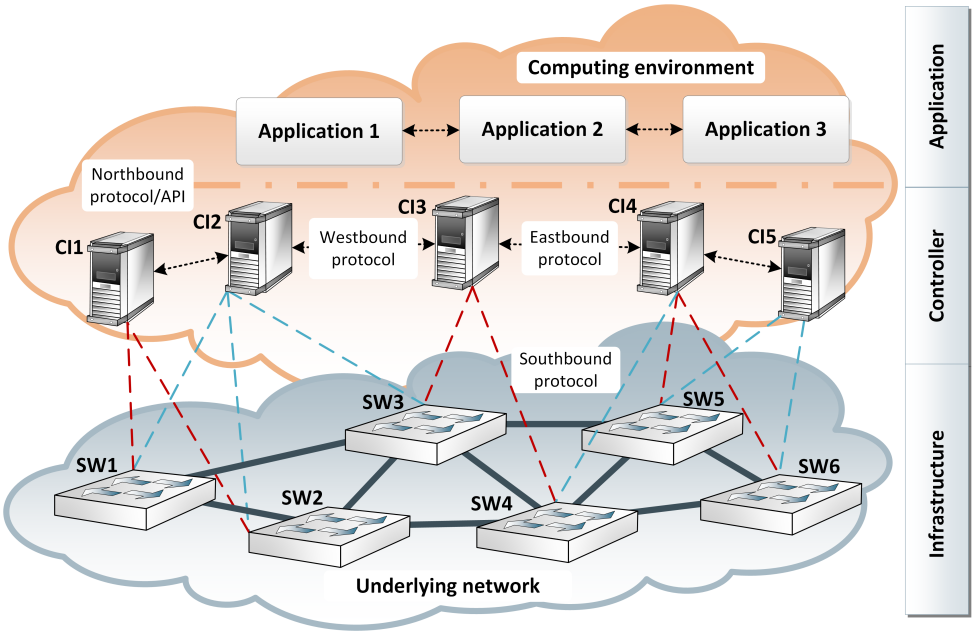


Figure 2.10: High-level view of the SDN architecture. The claret colored dashed links represent live TCP connections. On the contrary, light blue dashed links are backup TCP connections.

at least a partial availability of the management network.

The management network can be realized in two ways, either as in-band or out-of-band. While in-band management utilizes the same physical topology as payload traffic on the data plane, out-band management relies on dedicated physical network topology. Out-of-band management is supposed to perform better since it does not add any overhead to the shared topology and consequently, it does not consume any bandwidth at payload links. The out-of-band system is a preferred way of control even for the SG applications.

There are two main ways to make the controller highly available: replication and distribution [68]. The naive approach to controller replication is maintaining a running shadow copy of the controller. The shadow controller can instantly take over control in case the primary controller fails. Usually, a heartbeat mechanism is implemented between controller instances to check the availability. The replication is commonly a solution suitable for local areas and a limited number of flows.

The majority of controllers is usually built on some framework and their high-availability strategy comes from the framework's architecture. Although the need for the distributed control in SDN networks was mitigated by some projects like vertical-based ONIX, HyperFlow [69], [70] or horizontal-based Kadoo [71], other controllers implement distributed clustering based on strongly-consistent data-store replication and intra-controller synchronization between controller instances like the OpenDaylight controller [72] and ONOS [73].

The unclear situation prevails in the area of inter-controller communication protocol that has not been fully specified by any institution yet, even though some ideas were pub-

lished, for example SDNi [74]. Since the SDN concept is still evolving and the controller is a complex piece of software, one can predict that the range of controller projects will naturally sort out over the time.

2.3.2 OpenFlow protocol

The OF protocol[19] specification defines a structure, encoding and semantics of control messages, and it describes a way these messages are exchanged. The protocol can thus be viewed as a syntax notation for programming a packet processing pipeline [76]. The protocol was one of the first protocols that accelerated the idea of separating the control and data plane. Although the potential of the OF protocol is evident, it can also suffer from deployment issues as detailed by author of this thesis in [77].

Basically, the mechanism of packet forwarding in OpenFlow Capable Switch (OFCS) is based on matching[20] selected headers of passing packets[21] with flow entries[22] uploaded to the local FIB split into one or more flow tables. Each time the incoming packet matches a flow entry, the associated instructions are executed and counters[23] are adjusted. The flow entries are stored in flow tables with different priorities and specific timeouts; hard for absolute limits and soft timeout reflecting a flow lifespan. Matching rules are based on specific classifiers, i.e., packet header fields to be matched, and forms an ordered list of elements also called tuple. The dimension of the tuple expresses the rule granularity; the higher the dimension, the higher the granularity, and thus, a number of flow entries in flow tables. This statement is generally true, even though chaining flow tables allows to reduce number of flow entries preserving the same granularity.

Matching fields[24] are defined for packet header fields from the physical layer to the transport layer, if we consider the packet ingress port as a physical layer, metadata and other pipeline fields. The set of supported header fields has been growing with each released OF version supporting 40 fields in OF 1.5 [60]. The total number of flow entries is limited by the switch hardware, as most of them are aimed to be placed in Ternary

[19] The OF protocol was introduced in 2007 [59], and the first version of the protocol specification was approved in 2009 [75]. Lately, the development was adopted by the consortium called Open Networking Foundation (ONF).

[20] Matching means comparing the set of header fields and pipeline fields of a packet to the match fields of a flow entry [60].

[21] A packet is a series of bytes comprising a header, a payload and optionally a trailer, in that order, and treated as a unit for purposes of processing and forwarding. The default packet type is Ethernet, other packet types are also supported [60].

[22] A flow entry is an element in a flow table used to match and process packets. It contains a set of match fields for matching packets, a priority for matching precedence, a set of counters to track packets, and a set of instructions to apply [60].

[23] Counters are the main element of OpenFlow statistics and accumulated at various specific points of the pipeline, such as on a port or on a flow entry. Counters typically count the number of packets and bytes passing through an OpenFlow element, however other counters types are also defined [60].

[24] A matching field is a part of a flow entry against which a packet is matched [60].

Content-Addressable Memory (TCAM). Flow tables are implemented on the hardware level usually in either Random-Access Memory (RAM) or TCAM. In comparison with simple Content-Addressable Memory (CAM), TCAM provides a third state representing a "do not care" value used to implement wildcards in tuples. This state allows masks to be used for matching fields without the need of defining an exact value for each individual field. Although TCAM is very effective in matching, the cost and size (one TCAM cell consists of 16 transistors [78]) of the implementation is high and for that reason vendors often implement hash tables [79] for all types of lookups including ACL.

As can be seen from the processing diagram shown in Figure 2.11, an instruction is a cornerstone of the protocol. The instruction, introduced in OF 1.1, contains not only an output action, but it can realize a simple write to a set of actions, delete from the set, write metadata, jump to next table or use a meter[25]. An action to be used for header rewrite, push and pop for tagging, dispatch to egress port or more ports, dispatch to the regular switch pipeline, dispatch to controller, or pass the packet to be processed in group[26]. A group, stored in a group table, enables additional forwarding methods by grouping actions into buckets available for multiple flows. Depending on the group type, it is possible to effectively multicast or broadcast packets, implement load-balancing, or to realize a fail over in case of egress link failure. It should be noted, that not all features declared in the OF specification are defined as required, but as optional, which limits the real usability of the protocol.

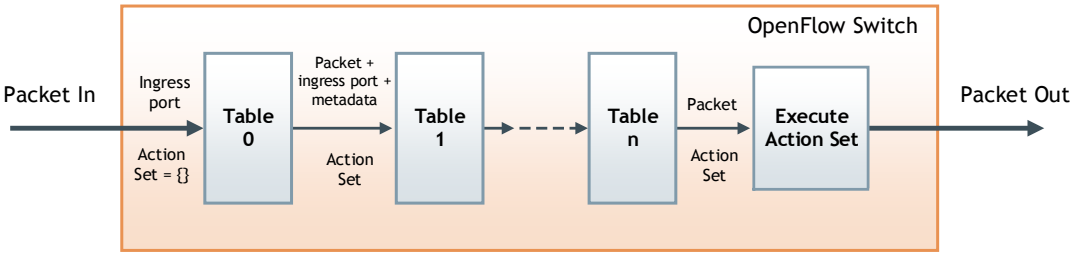


Figure 2.11: Packet processing model employing more flow tables.

The controller can add, update, invalidate, and delete flow entries in flow tables, both reactively and proactively [60]. In the proactive case, flow entries are uploaded to the network on the basis of network and service requirements before the particular flow appears. Using the reactive approach, flow rules are installed to the network in the moment an unknown packet is dispatched to the controller and subsequently analyzed to create a new flow entry. This behavior depends on the requested miss flow entry policy, as

[25] A meter is a switch element that can measure and control the rate of packets. The meter triggers a meter band if the packet rate or byte rate passing through the meter exceeds a predefined threshold [60].

[26] Group is a list of action buckets and some means of choosing one or more of those buckets to apply on a per-packet basis [60].

the frame may be dropped, forwarded to the controller or it may continue to the normal switch pipeline.

The detection of the loss of switch-controller connection is not based on any TCP features but must be made on the application layer by exchanging OF ECHO messages. There is no specification of timer values for exchanging this type of messages. This means that the controller has to ensure a proper check of connection to meet QoS requirements of services utilizing the network.

2.4 Network topologies

As the application area and tools are clarified, we continue to the topic of network topologies that are important from the optimization perspective. The optimization techniques are different, and thus, they can be sensitive to structural features of the particular underlying graph topology. The following Section briefly introduces a selection of topologies, both practically deployed and artificially created, to limit our research to a useful set of problem instances. As the graph composed of nodes and links connected in a non-trivial topology is a complex network, the term network and graph is for our purposes used interchangeably in the following text.

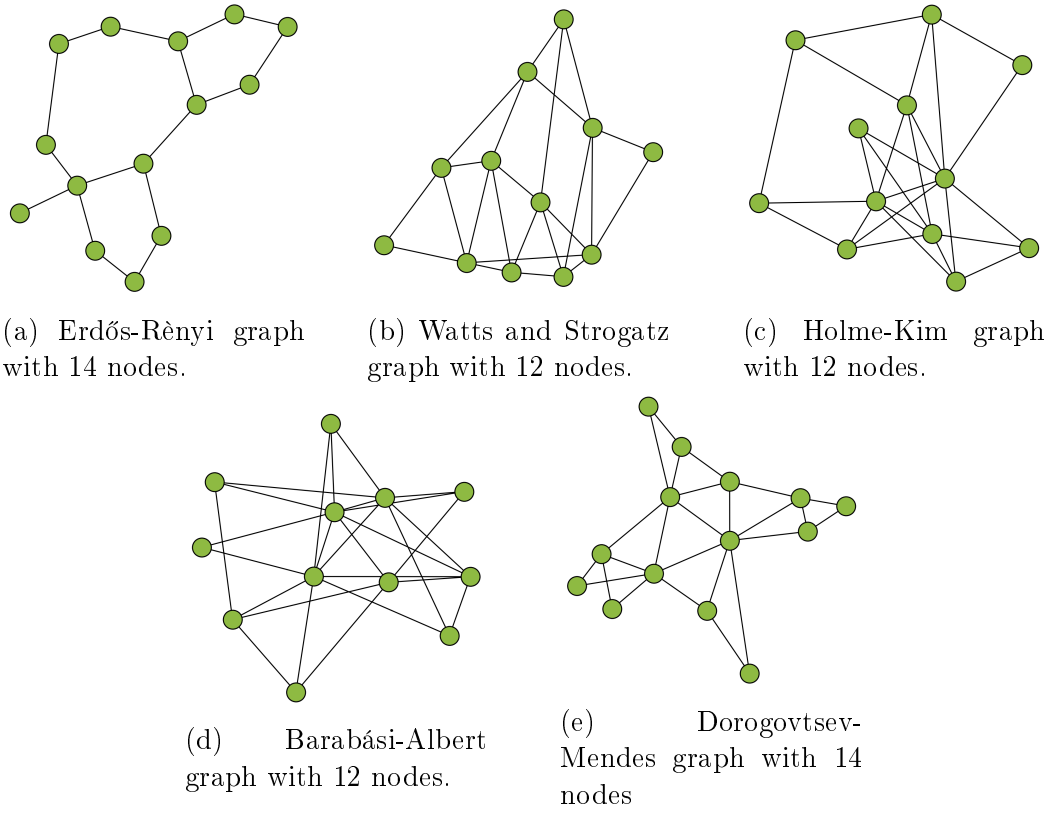
2.4.1 Random graphs

At first, four different random models were chosen with respect to a sufficient variability needed for the initial evaluation of the proposed algorithms. The assumption is that larger graphs containing a high number of links and nodes interconnected with high entropy provide a better substrate for the optimization. As the random graphs pose the required diversity in contrast to the regular LAN topologies, these are more feasible to better understand the features of the proposed algorithms when numerically evaluated. Topological characteristics [80] for a particular graph used for the evaluation are presented in the relevant chapters.

Random graph topologies were generated under different conditions following the Erdős-Rènyi model [81], Watts and Strogatz model as a small-world model[27] [83], Barabási-Albert model [84] and a planar Dorogovtsev-Mendes model [85] as scale-free[28] representatives. A sample of generated graphs is shown in Figure 2.12.

[27] Small-world network is an analogy to the small-world phenomenon, know also as six-degrees of separation. The small-world network refers to an ensemble of networks in which the mean shortest-path between nodes increases sufficiently slowly as a function of the number of nodes in the network [82].

[28] In scale-free networks, the degree nodal distribution follows a power-law.



(a) Erdős-Rènyi graph with 14 nodes.

(b) Watts and Strogatz graph with 12 nodes.

(c) Holme-Kim graph with 12 nodes.

(d) Barabási-Albert graph with 12 nodes.

(e) Dorogovtsev-Mendes graph with 14 nodes

Figure 2.12: Example artificial topologies generated according to random graph models.

2.4.2 Regular topologies

Next, regular network topologies deployed in LANs were generetad. While the typical network topology in SAS is intended to follow the structure of the Station and Process bus, as shown in Figure 2.8, the real installations are not restricted to the same physical infrastructure. The logically structured topology can be converged to one physical topology to effectively manage, control, and utilize transmission means while keeping a high level of network robustness.

The network robustness can be described and quantified by many metrics [86]. As is summarized in [86], a traditional network robustness definition, that comes from the graph theory, is as following: *a network is robust if disconnecting components is difficult.* A more contemporary definition [87]: *robustness is the ability of a network to maintain its total throughput under node and link removal.* The author of this thesis researched robustness metrics in [88] and proposed a novel centrality measure Quality of Alternative Paths (QAP). QAP centrality quantifies node surroundings and can be with an advantage utilized in algorithms to indicate more robust paths in the network.

In the context of the SAS, one of the most important parts of the substation protection functions is to keep GOOSE communication functional between IEDs under any condition. This means guaranteeing a host-to-host reachability through one graph component,

also known as All Terminal Reliability (ATR) [89] even under massive failure of network devices.

Beside fundamental topologies, bus, tree, linear, and single star, which do not provide any level of redundancy and are prone to network failure, more redundant topologies were identified to avoid Single Point of Failure (SPOF) to improve network robustness. Among the identified topologies are mainly extended star-ring topologies with horizontal interconnections, i.e., hybrid topologies, inspired in campus networks, industrial ring network and date center networks.

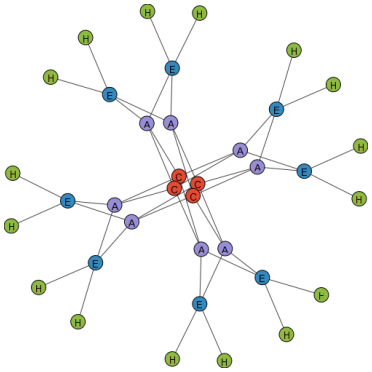
Since the SA network performance requirements as bandwidth, latency and reliability, are the same for data centers, two proposed network topologies have their origin in this area. In recent years with the pressure on cost and efficiency, especially in data centers, one can see the inspiration of data network architectures in traditional non-blocking networks proposed by Charles Clos [90]. These architectures are based on commodity switches such as the Fat Tree proposed by Al-fares et al. in [91]. Formally, Fat Tree topologies are *k-ary n-trees* but sometimes are also termed as Leaf and Spine topologies. Another traditional topology in data centers is three-tier architecture [92]. The inspiration in data center networks based on commodity switches is reasonable as research shows in [93] where the authors observed that the data center network exhibits high reliability with more than 99.9999 % of availability for about 80 % of the links and for about 60 % of the devices in the network.

All topologies depicted in Figure 2.13 are constructed with a high level of node and link redundancy as these are viable for the optimization algorithms. Most topologies are comprised of three levels, and core switches are always interconnected. For illustrative purposes, graphs are drawn with end hosts denoted as H, representing IEDs. While all topologies are resilient to SPOF for internal communication on the level of interconnecting nodes, hosts are connected without any connection redundancy. Graph nodes represent Ethernet switches on topology levels A (Access, Aggregation), E (Edge), D (Distribution) and C (Core). The terminology differs depending on the area where such network topology is to be deployed.

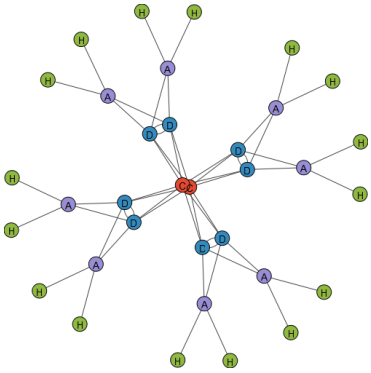
2.5 Steiner tree problem

The following section summarizes the categorization of the SP, different problem definitions, and algorithms.

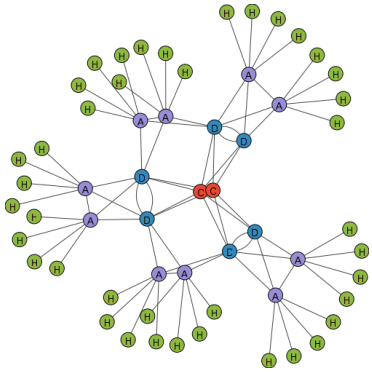
Since the basic SP in networks may not be clear, Figure 2.14 shows the essential difference to Shortest Path Tree (SPT). This example should clarify the SP before we step deeper into the problem classification.



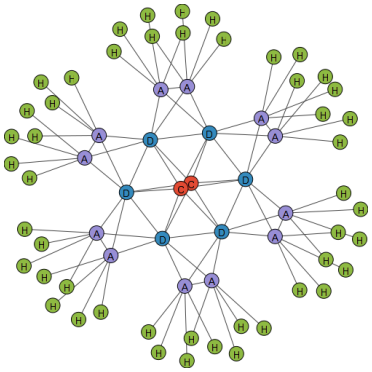
(a) Fat Tree data center topology with $k = 4$ *ft*.



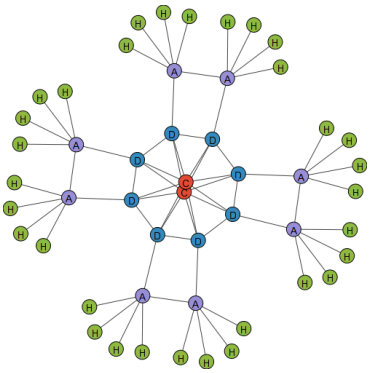
(b) Three-tier data center topology with 2 core switches – *ttg*.



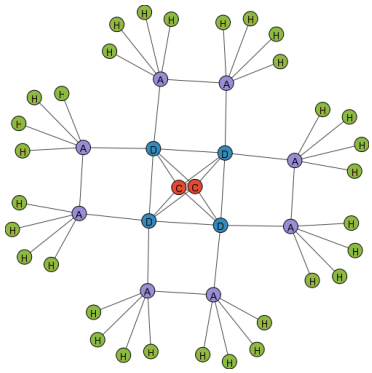
(c) Campus tree topology in pyramid order – *ctg9*.



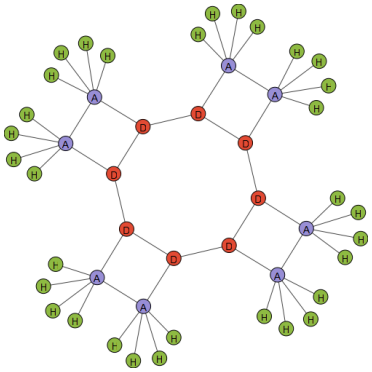
(d) Redundant Campus tree topology – *ctg10*.



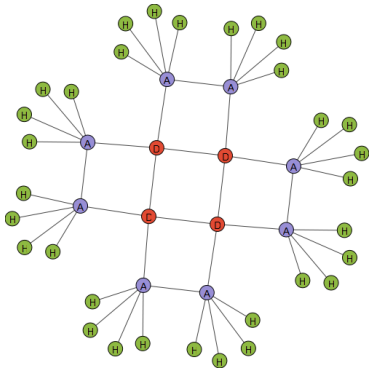
(e) Campus ring topology interconnected via dual ring nodes with a ring link – *crg35*.



(f) Campus ring topology interconnected via a shared interconnection node – *crg36*.



(g) Ring topology interconnected via dual ring nodes with a ring link – *rrg47*.



(h) Ring topology interconnected via a shared interconnection node – *rrg48*.

Figure 2.13: Regular hybrid topologies with redundant connections.

2.5.1 Problem classification

As already mentioned in 1.2, finding the optimal multicast tree can be viewed as SP. The SP in networks is $\mathcal{NP}$-complete problem. This means that it is possible to verify in polynomial time, thus with complexity $\mathcal{O}(n^k), k \in \mathbb{N}$, whether a given solution x from

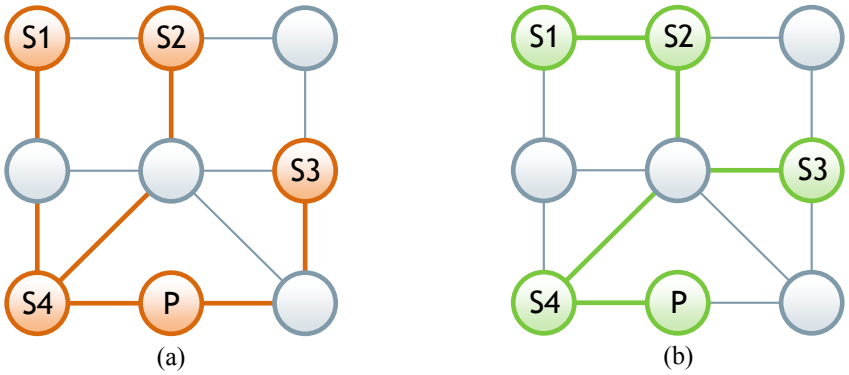


Figure 2.14: Comparison of (a) shortest path tree (total cost = 7, max path length = 3, avg. path length = 2.25) and (b) minimum Steiner tree (total cost = 5, max path length = 4, avg. path length = 2.27) for the same multicast group and the same multicast source. Unit lengths and unit costs are assigned to all links. The multicast group $G = \{S_1, S_2, S_3, S_4\}$. Node P is the multicast source [94].

a solution space is a valid solution. $\mathcal{NP}$-completeness has been originally proven for minimum Steiner Tree in Graphs by Karp in [95]. Expanding the original SP for QoS constraints like bandwidth or delay, the complexity class remains in $\mathcal{NP}$-complete as is shown in [96].

The minimum Steiner Tree in networks can be classified even more thoroughly. It is an optimization problem and as such it falls into the $\mathcal{NPO}$ class[29]. Problems in this class satisfies the following conditions: solution is polynomially bounded in the length of instance, whether x is a solution of the instance is decidable question in polynomial time and the measure of solution quality is computable in polynomial time. More specifically, minimum SP is $\mathcal{APX}$-complete. $\mathcal{APX}$ is a set of all optimization problems in $\mathcal{NPO}$ which admit a polynomial time approximation algorithm (heuristic) with performance ratio[30] $\rho \geq 1$ [10]. Since in most cases the optimal solution is unknown for the particular instance, other approaches are used for comparing heuristics such when the optimal solution is considered the best reached [97].

If only one multicast source exists in the group, then the multicast tree is referred to as a source-based tree or one-to-many tree. If there are more terminating nodes, which also act as multicast sources, then we speak about shared trees or many-to-many trees [19], [98]. Similarly, the multicast routing problem can be divided into two versions: static and dynamic. In the static version, the algorithm deals with the initial creation of the multicast connection topology. The source of the multicast and members of the multicast group are known in advance. On the contrary, in the dynamic version, group members

[29] An $\mathcal{NP}$-optimization problem ($\mathcal{NPO}$) is a combinatorial optimization problem.

[30] The performance ratio ρ is a measure of the quality of approximation algorithms defined as the ratio of the approximative solution to the optimal solution.

can freely join or leave the multicast group during the group's lifetime. From the nature of the proposed SDN control and the presented multicast problem, where group members are known in advance and sources produce nearly constant data flows, **the thesis deals with a centralized, static and source-based tree**.

2.5.2 Published algorithms

Published heuristics consider different objectives when optimizing the multicast tree, such as path delay, total cost of the tree or maximal congestion. These objectives make SP optimization even harder. In the case of the multi-tree BDLDV multicast problem, the constraints are clear. In contrast to the typical goal of the SP optimization, where the minimum sum of costs for links used in the multicast tree is searched, the goal of the BDLDV is to minimize delay-variation of delivered multicast messages. Even though the goal is slightly different, problem complexity remains in the same class, and hence, it is subject to approximative algorithms. The minimum SP assumes a fundamental condition $3 \leq |s \cup R| \leq |V| - 1$, where s is a source node, R is a set of receivers and $|V|$ is the total number of graph nodes. In the case where the number of multicast members is close to the total number of nodes, SP in networks can be simplified to the MST problem, for which the algorithm with polynomial time complexity is the known Jarník-Prim algorithm [99].

The vast majority of published algorithms are source-specific focusing on minimization of tree cost often considering at least one constrain which is usually a maximum transmission delay required by applications. Almost all basic heuristics use or come out of two basic ideas known from the MST problem [97].

Insertion The initial tree $T(s, \emptyset)$ consists of a single basic vertex. Then, expand T to a feasible solution by adding all basic nodes, for example through the computation of at most $|R|$ of the shortest paths. This approach corresponds with the Jarník-Prim algorithm.

Component connecting The initial solution is a partial tree $T(s \cup R, \emptyset)$ consisting of basic nodes. Then, expand T to a feasible solution by a computation of shortest paths linking all components.

Basic heuristics also employ hybrid methods based on the ideas presented above to reach better results. The first method directly implies a simple heuristic called *Shortest paths*, where the heuristic constructs a shortest path from an initial multicast publisher to all multicast subscribers using an arbitrary shortest path algorithm. The second simple heuristic is based on *MST and pruning*. Firstly, it is constructed a MST denoted T of

graph G. Secondly, each tree leaf from T, being a Steiner node[31], and incident edges are iteratively removed in a loop, until the minimum sufficient tree remains.

One of the most mentioned minimum SP heuristics on graphs was proposed by Kou et al. in [100] in 1981, often referred by the authors Kou, Markowsky and Berman (KMB). Mentioning this algorithm has a practical interest, since it has a performance guarantee of at most twice the size of the optimum source-based Steiner tree. The steps of the KMB heuristic are sketched in Algorithm 1.

Algorithm 1 Minimum spanning tree heuristic KMB for SP on graphs

Construct a complete graph $K(R, E)$ where the set of nodes is R.
Let the distance $d(i, j), i, j \in R$ be the shortest path from i to j in G.
Find an MST T of K.
Replace each edge (i, j) in T by the path from i to j in G. Let the resulting graph be T'
repeat
  $r \leftarrow false$
  if there is a leaf $w \in \hat{T}$ which is not in R **then**
    Remove w from $\hat{T}$
    $r \leftarrow true$
until not r

As the SP is one of the fundamental optimization problems, a plenty advanced algorithms solving the constrained minimum SP have been proposed during the times. In many cases, based on variation of Bellman-Ford's or Dijkstra's algorithms. Generally, these algorithms are focused only on source-based trees with one multicast source considering different QoS criteria. From the perspective of our LDV problem, two kinds of problems related to the delay variation were historically studied: Delay and Delay Variation-Bounded Multicast Tree (DVBMT) and Delay and Delay Variation-Bounded Steiner Tree (DVBST). The latter additionally considers tree cost for an objective in optimization formulation [101]. Rouskas and Baldine defined the DVBMT in [96], where they proposed a Delay Variation Multicast Algorithm (DVMA) that has a series of successors improving time complexity. The DVBST problem became a main topic of various algorithms [101]–[103]. A summary of the important multicast routing algorithms recognized in [98] is given in Table 2.2. All listed algorithms are static and central giving more accurate results with the only exception being MOSPF. To run such heuristics, the node has to hold all relevant state information about the network which means that the algorithm usability is limited by the device performance.

While source-specific multicast routing has been extensively researched, little attention has been paid to a many-to-many multicast routing. This multi-source multicast can be

[31] Steiner node is a node that is neither publisher nor receiver node but participates on the message transmission.

Table 2.2: Source-specific multicast routing algorithms (Δ = delay constraint, m = multicast group size)

Algorithm	QoS	Complexity	Reference
Unconstrained shortest-path heuristics	No	$\mathcal{P}$	
◦ MOSPF	No	$\mathcal{O}(\|V^2\|)$	[20]
MST (Prim's)	No	$\mathcal{O}(\|V^2\|)$	
Unconstrained minimum Steiner tree	No	$\mathcal{NP}$-complete	
◦ KMB heuristic	No	$\mathcal{O}(m\|V^2\|)$	[100]
◦ TM Heuristic	No	$\mathcal{O}(m\|V^2\|)$	[104]
Delay constrained shortest path	Yes	$\mathcal{NP}$-complete	
◦ Constrained Bellman–Ford (CBF)	Yes	Exponential	[105]
◦ Constrained Dijkstra heuristic (CDKS)	Yes	$\mathcal{O}(\|V^2\|)$	[94]
◦ Bounded shortest multicast (BSMA)	Yes	$\mathcal{O}(k\|V^3\|log(\|V\|))$	[106]
Delay constrained minimum Steiner tree	Yes	$\mathcal{NP}$-complete	
◦ KPP heuristic	Yes	$\mathcal{O}(\Delta\|V^3\|)$	[107]
DVMA	Yes	$\mathcal{NP}$-complete	[96]
Delay and bandwidth constrained	Yes	$\mathcal{P}$	
◦ Shortest-widest heuristic	Yes	$\mathcal{O}(\|V^3\|)$	[108]
◦ Widest-shortest heuristic	Yes	$\mathcal{O}(\|V^2\|)$	[109]
◦ Bw, hops, extra cap. (BHE) heur.	Yes	$\mathcal{O}(\|V^2\|)$	[109]
◦ Nearest destination first (NDF)	Yes	$\mathcal{O}(\|V^3\|))$	[109]
◦ MBDC	Yes	$\mathcal{O}(\|V^2\|log(\|V\|))$	[98]

distributed on a single shared tree or Multiple Shared Multicast Trees (MSMT). A single shared tree clearly has a disadvantage in higher propagation delays than more source-specific trees. On the other hand multiple shared multicast trees can benefit from a properly placed center node or nodes, in terms of PIM-SM called Rendezvous Point (RP). Each multicast source sends packets toward each of the RPs, but a receiver joins only a tree of single RP. This approach results in each RP having its own shared tree that spans all sources but only a subset of destinations [94]. Properly placed center nodes should bring more stable latencies which is an important advantage from the perspective of the LDV problem. There are few algorithms dealing with the MSMT problem on minimizing the number of center nodes which is also $\mathcal{NP}$-complete. Salama proposes several algorithms to solve the MSMT problem in his PhD thesis [94]. The most interesting is GREEDY heuristic which has two phases. In the first phase, the MSMT problem is transformed to a *Set Covering problem*, and then, in the second phase, it is used the *Greedy* algorithm to find an optimal solution. The *Greedy* algorithm does not assure a globally optimal solution since it decides only according to the locally best solution in every stage of the evaluation.

The heuristics mentioned above are, in most cases, techniques adapted from MST-based algorithms which are sort of iterative methods generating a sequence of improving approximative solutions. However, these are not the only heuristics. Another possibility

is to apply metaheuristics[32] using stochastic optimization methods which generate and use random variables to find an approximative solution. The randomness and proper heuristic initiation enables locally optimal solutions to be overcome and move closer to the globally optimal solution. Even these methods do not ensure the optimal solution discovery, but they may provide a sufficient solution in a reasonably short time. Many papers have been published on constrained minimum SP, for example, *Simulated Annealing* [111], *Tabu Search* [112], *Greedy Randomized Adaptive Search Procedure* [113], or evolutionary algorithms such as GA [114], [115]. Since GA is in the area of our interest, further descriptions focused on GA and its basic principles and phases are detailed in Chapter 5.

As Terzian shows in her thesis [116], multicast forwarding is a persisting research topic. Although Terzian works with 2D mesh and torus networks, the most interesting is, from our perspective, the proposed multi-core DVBMT that effectively improves cases where single-core based algorithms fail to fulfill defined constraints.

Currently, the authors focus on more complex variations of the multicast forwarding problem considering multiple objectives with multiple constraints. Published algorithms address the increasing level of problem complexity by combined metaheuristics. Recently, Xu and Qu presented in [117] an extensive survey of metaheuristics together with a proposal of multiobjective simulated annealing based on a genetic local search algorithm that represents the combined approach. Concurrently, the multicast forwarding problem is drifting from the traditional network layers up to the application layer, since the trend of overlay networks and applications not relying on lower layers is more apparent than ever as published in [118] by Lin et al. In the context of methods used in this thesis, Park et al. published an ILP formulation of a multi-QoS DVBST variant used for multicast routing in sparse-splitting optical networks [119]. The authors claim the ILP is widely used to solve multicast routing optimization problems in optical networks.

[32] A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms [110].

Amazing. This is what was missing from the dig at Giza. This is how they controlled it. It took us 15 years and three supercomputers to MacGyver a system for the gate on Earth.

Samantha Carter

3

Latency on Switched Ethernet Networks

Since total transfer time is the critical constraint for defined optimization problems, we need to understand what causes delays in switched Ethernet networks. This chapter describes specific latency sources participating in the message transmission as defined in Section 2.2.3, and proposes a methodology for the measurement of switch fabric[1] latency.

3.1 Latency sources

Although some latency sources can be determined analytically, some have to be measured or can be estimated only for particular loads in the network. In the switched Ethernet networks, we can identify several latency sources: serialization and deserialization latency, store-and-forward latency, wireline latency, switch fabric latency, and queuing latency.

The delay introduced by frame serialization on a sender's egress port is equal to the deserialization delay on a receiver's ingress port that happens simultaneously though only when time-shifted by the wireline latency. In the context of the switched Ethernet network, the deserialization delay is synonymous with store-and-forward latency. All these three latencies can then be considered as a single latency, and thus, the following text focuses only to store-and-forward latency.

[1] A silicon crossbar switch chip remotely resembles a woven piece of fabric, and hence, it became known as a "switch fabric", "silicon fabric", or simply a "fabric".

3.1.1 Store-and-forward latency

The first source is given by the operation principle of an Ethernet switch and how it processes the incoming frames. In the fastest method *Cut-through*, the frame does not undergo any checks, as the checksum is at the end of the frame. It is sent straight after the destination Media Access Control (MAC) address is read and the output port is determined. Although this approach is lightning fast, the transmission reliability is overall decreased. The *Cut-through* method can be applied only under specific conditions and the forwarding decision relies on the switch on a frame-by-frame basis. It can occur only between interfaces operating at the same data rate, never from a slower to faster interface, and only when the egress port is in an idle state.

In other cases, the de facto standard in LANs called the *Store-and-forward* method has to be employed. The whole incoming frame is stored in an internal memory at first, and only after validation, it passes toward the destination port. For SG applications, the only acceptable method is the *Store-and-forward* method. As expression (3.1) states, latency introduced by the store-and-forward method is proportional to frame length and inversely proportional to the net bit rate.

$$t_{saf} = \frac{l_f}{R_{nb}} \tag{3.1}$$

where l_f is the length of the incoming frame [bit] and R_{nb} is the channel net bit rate [bps]. For example, latency introduced by the *Store-and-forward* method in case of a 1518 byte (12144 bits) length frame on a 100Mbps channel is 121.44 µs, and the minimum length frame of 64 bytes (512 bits) on 1Gbps channel entails only a 0.5 µs delay.

3.1.2 Wireline latency

The second element involved in total transfer latency is a link propagation delay taken as a piece of information, a bit, to traverse the point-to-point connection along a cable. The signal propagation delay can be estimated as shown in expression (3.2), where l_c is the cable length, c represents the speed of light ($3 \times 10^8\,\mathrm{m\,s^{-1}}$) and Nominal Velocity of Propagation (NVP)[2]. NVP expresses the speed signals traveling in the cable relative to the speed of light in the vacuum.

$$t_{sp} = \frac{l_c}{NVP \cdot c} \tag{3.2}$$

Even though wireline latency is considered to be negligible in LANs, because of the maximum length of links, the advent of networks operating on data rates higher than

[2]NVP depends on the transmission medium. Air is close to a vacuum with $NVP \approx 0.99$, a standard optical fiber G.652 has $NVP \approx 0.68$, a twisted pair cable $NVP \approx 0.67$, and coaxial cable $NVP \approx 0.82$.

1 Gbps, where switch fabric latency approaches close to 1 µs, the propagation delay can be more significant. Let's consider a 100m long optic fiber. Wireline latency is, in such a case, attacking the level of 500 ns which is a half of the switch fabric latency and not an order of magnitude as it used to be at lower data rates.

3.1.3 Switch fabric latency

The switch fabric is generally a silicon crossbar switch that realizes the interconnection of the ingress and egress port and allows the frame to be properly forwarded. The design of the fabric can vary depending on the switch architecture but usually it is accompanied by a controller that makes the decision where to forward the frame, and thus, how to set up the crossbar. While the very basic Ethernet switch works in this way, the modern switches above the Small Office Home Office (SOHO) market implements a tremendous amount of advanced functions. For this reason, it is almost impossible to separate pure switch fabric latency from the overall packet processing and functional logic implemented inside the particular Ethernet switch. As the results in Section 3.2.4 show, switch fabric latency typically starts around the lower tens of microseconds at 10Mbps data rates and goes down to hundreds of nanoseconds at high end data center switches at 10 Gbps. The architectural limitations of the switch fabric latency measurement and comparison of different level switches are detailed in Sections 3.2.2 and 3.2.4.

3.1.4 Queuing latency

The most unpredictable and non-deterministic part of end-to-end delay evaluation in switched Ethernet networks is latency introduced by queues, i.e., buffers, in switches. Irrespective to a particular switch architecture, queues in conjunction with the Store-and-forward mechanism are an essential part of the Ethernet design to avoid collisions on the medium. In the time of a full-duplex switched Ethernet, the main purpose is to prevent congestion on an egress port, usually when there is a burst of frames from a higher data rate interface to a lower data rate interface or when more flows are directed to the same output.

Generally, Ethernet switches interconnected in such cases form a network of queues [120] where queuing is the factor significantly contributing to the frame transmission latency. However, each switch would evince an excessive frame drop rate without queuing every time an egress port is occupied by another frame transmission. Queuing latency can be partly mitigated in cases when a switch implements some of the priority forwarding mechanisms when taking Class of Service (CoS) of the particular traffic into consideration. Nonetheless, frames with common CoS are usually handled in the same way, meaning the

best-effort, and cannot guarantee QoS since they are queued relative one to another. Moreover, there can always be a frame with lower CoS already transmitted on the egress port.

Calculating the worst case latency introduced by the queuing is very challenging, it requires an exact knowledge of all traffic generated in the network: frame dispatch times, CoSs, time distributions and rates of the frames. Since the great level of uncertainty of the frame ordering in queues, the calculation of the queuing latency is commonly attempted as a simple average estimation based on the network traffic load. In such case, it is assumed that the likehood of a frame already enqueued is proportional to the network load [121].

$$t_q = L_n \cdot t_{saf_{max}} \tag{3.3}$$

where L_n is dimensionless variable reflecting a relative network load of the total transmission capacity, and $t_{saf_{max}}$ is a store-and-forward latency caused by queuing the maximum-size frame (1518 byte, neglecting jumbo frames). For example, consider a network load approximately 25 % on 1Gbps channel, than, the queuing latency obtained for 1518-byte long frame is 3.036 µs. In extreme cases, when an excessive traffic, i.e., aggregated traffic from several ingress ports, is forwarded to an egress port is constantly greater than the output data rate frames overflowing the already filled queue are dropped. From the queuing theory for the M/M/1 model, we can state the queue stability condition is no met in such case, thus, the intensity of the arrival process (traffic) $\rho = \frac{\lambda}{\mu} \geq 1$ Erlang, where λ is a mean rate of arrival requests and μ describes a mean rate of service time [120].

3.1.5 End-to-end delay

The above mentioned sources of latency repeatedly contribute at each hop between switches to the end-to-end delay on the way from the frame producer to its terminal consumer. The total transmission delay can be generally calculated as in expression 3.4.

$$t_{e2e} = \sum_{i}^{N_{sw}} (t_{saf_i} + t_{sw_i} + t_{q_i} + t_{sp_i}) + t_{saf_r} + t_{sp_r} \tag{3.4}$$

where N_{sw} express a number of switches along the path from the producer to the consumer, t_{saf_i} is store-and-forward latency on i-th switch ingress port, t_{q_i} is queuing latency on i-th switch egress port, t_{sp_i} is wireline latency on i-th network segment, and t_{sw_i} is switch fabric latency of i-th switch in a row. Considering the whole transmission chain, the latency introduced on the last hop to the consumer is expressed as store-and-forward latency t_{saf_r} at the consumer's ingress port and a wireline latency t_{sp_r} from the

last switch to the consumer.

3.2 Measurement of switch fabric latency

The example of stringent transfer time requirements in SASs shows how important is to be aware of all network parameters when designing the network topology. From this perspective, an incomplete dataset is the switch fabric latency. Although vendors mention the switch fabric latency in datasheets, numbers are mostly carried out under undefined conditions, that is not sufficient for the implementation in industrial networks. For this reason, we decided to design a new measurement methodology allowing us to verify information provided by vendors.

The key objective was to develop and evaluate the measurement methodology to determine the switch fabric latency for data rates up to 10 Gbps. The measurement was based on a zero-load approach to avoid any buffering delay. Although the methodology was firstly published in [122], the core of the following section was published by the author of this thesis in [123]. The proposed methodology is enhanced and was extensively evaluated on switches supporting data rates up to 10GBase including OpenFlow switches.

The following text is organized as follows. Section 3.2.1 presents related works and standards. Sections 3.2.2 and 3.2.3 present measurement limits and measurement methodology itself. The last section, Section 3.2.4 describes a number of experimental measurements carried out in our laboratory with the aim to verify and demonstrate applicability of the proposed methodology.

3.2.1 Related works

The perennial weakness of the latency measurement is a source-receiver time synchronization, i.e. between a source of the measurement signal and the receiver that is able to compute the latency. Published papers dealing with the measurement of switch fabric latency exploit different approaches. The synchronization can be achieved by a time synchronization protocol to determine the latency, e.g. Network Time Protocol (NTP) as suggested by Loeser et al. in [124]. However, NTP is not suitable for precision applications, as it is not accurate enough. In order to achieve better synchronization, IEEE 1588 is a more satisfactory option [125]. Apart of the network protocols, an option is to use internally synchronized specialized card that provides a high-precise frame timestamping and implicitly measure passing frames in a loopback setup as is suggested by Ingram et al. in [13]. Both approaches require additional specialized hardware dependent on the transmission technology or a dedicated synchronization protocol.

The next option is to measure the latency directly by means of special data frames called CFrames, as suggested in [126]. The authors of this paper suggest to use a flow CFrames passing through the internal switch fabric and to measure latency between the ingress and the egress port directly at a switch backplane. Since the integrated design of most switches prevents the access to the switch fabric this approach does not address real use-cases.

As we deal with measurements in switched Ethernet networks, the fundamental standard is IEEE 802.3 [127]. It covers all variants up to 100Gbps Ethernet and describes the encapsulation process down to Physical layer (L1).

The elementary description of the switch fabric latency is based on RFC 1242, which defines the latency of the store-and-forward devices [128]. According to the Request for Comments (RFC), the switch fabric latency, or in other words processing time of the passing frame, is determined as the time interval starting when the last bit of the input frame reaches the input port and ending when the first bit of the output frame is seen on the output port. This approach is typically called Last In First Out (LIFO). An important document related to the measurement methodology is as well RFC 2544 [129]. This RFC defines, inter alia, the time intervals necessary between individual measurements as well as frame lengths needed for the measurement, and a wide range of specialized measuring instruments implement this recommendation in their basis.

A fundamental document for the methodology accuracy evaluation is a technical report by Joint Committee for Guides in Metrology [130] specifying the calculation of measurement uncertainty.

3.2.2 Switch architecture and measurement limits

The internal architecture of Ethernet switches differ significantly, as it constitutes vendor's know-how. However, there are in most cases common architectural components fulfilling the primary functionality. Generally, the switch can be seen from two perspectives. The first comes from the arrangement of physical components in the device. The second is the switching logic itself, i.e., how the switch queues frames, schedules forwarding and implements these functions into a memory.

From the hardware point of view shown in Figure 3.1, the switch is composed of line cards, Central Processing Unit (CPU), various memories for storing the FIB and the switch fabric. The fabric is usually implemented as an Application Specific Integrated Circuit (ASIC). All components are connected by an internal bus situated on the switch backplane. The line card contains at least one interface for signal processing at the Physical layer (PHY) and MAC. To speed up the forwarding, it can also contain a local FIB and a fabric ASIC if the line card hosts more ports. The architecture of modular

switches and large enterprise switches is different in both terms of backplane design and line card construction. They are usually supplemented by additional CPUs and memories.

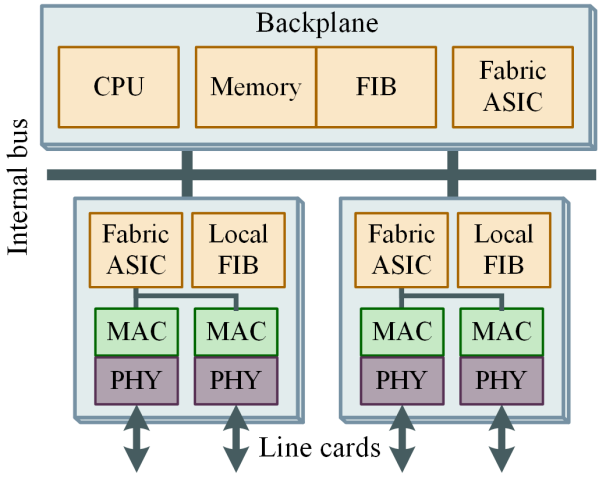


Figure 3.1: Physical arrangement of the components in a general switch.

The architecture of the switch heavily depends on the way how a frame is internally processed and available memory utilized. Current switches mostly implement the shared memory method, which is frequently applied architecture due to its efficient utilization of resources and the best delay-to-throughput ratio. In general, the internal architecture can be classified using the queuing strategy a an input queuing, an output queuing, or a combination of output and input queuing.

Although the most effective strategy in terms of the throughput utilization is based solely on the output queuing, it poses a great demands on the internal switch bus and switch fabric. It must run N times of the highest line speed to serve all N inputs without blocking. Such approach is impractical due to the limit on size and capacity of the switch. Since the output queuing architecture is economically demanding and architectures based on input queuing suffers by Head of Line (HOL) blocking, different strategies were proposed in the past.

Typically, vendors implements two approaches. The first one is Combined Input and Output Queuing (CIOQ) system, where each frame is buffered twice; before the fabric in Input Queuing (IQ), or Virtual Output Queuing (VOQ) [131], and before egress port in Output Queuing (OQ). The second is VOQ system, where the incoming frame is buffered only once; before the switch fabric, and it is pulled out of the queue when the all way down to egress port is available. The VOQ solution prevents the head of the queue from being blocked. Both strategies require a complex management; thus, it is expectable that the management can add a not negligible time component to the overall switch fabric latency. Besides the queuing management, the switch fabric latency is influenced by mechanisms how forwarding decisions are evaluated and policy enforced. Generally, more decisions

left to CAMs and TCAMs, detailed in Section 2.3.2, shorter time needed to process the frame.

Accordingly, the overall processing time of the frame transmission between the ingress and egress port is composed of several independent delays. The minimum measurable switch fabric latency within the common architecture can be estimated as (3.5)

$$t_{sw} = t_{iq} + t_{sf} + t_{oq} + 2t_{lc} \tag{3.5}$$

where t_{sw} stands for the total switch fabric latency, t_{lc} represents the line card delay, i.e., the processing time of the frame passing between layers and the time needed to transfer the frame via the internal bus to the switch backplane, t_{sf} is the actual delay on the switch fabric, t_{iq} is the input queue delay, e.g., and t_{oq} represents the output queue delay. The line card delay does not involve an input buffering delay eliminated by the LIFO measurement approach. As one can see, measurable t_{sw} is susceptible to various internal processes. The proposed measurement methodology minimizes their impact, and thus, t_{sw} is considered to be an acceptable estimate of the real switch fabric latency.

3.2.3 Measurement methodology

As mentioned in 3.2.1, the fundamental principle of the measurement methodology is based on the LIFO method. The methodology was designed firstly to measure only 10Base-T Ethernet switch fabric latency, and later it was enhanced for additional Ethernet revisions. All measurements are performed at the 10Base-T Ethernet channel.

First measurement scenario builds on top of the Manchester encoding at 10Base-T channel which is not burdened by any broadcasting during the idle state between particular transmissions. Since the channel remains silent between transmissions, the passing test frame can be identified unambiguously. Other Ethernet variants at higher data rates keep uninterrupted idle signal on the transmission channel to preserve the sender-receiver bit synchronization. Thus, it is not possible to determine the head and tail of the passing test frame at the L1 without any signal decoding. Since the encoding schema is not the same for all Ethernet technologies and it requires high performance packet analyzer, this type of measurement is challenging and therefore not preferred. To apply the proposed methodology, only single two-channel oscilloscope is required.

The test traffic consists of Internet Control Message Protocol (ICMP) packets. The data stream of subsequent test packets is generated by the sender using the ping application. This application is sufficient for measuring purpose because it allows to adjust packet length and time spacing between individual packets. All unnecessary features potentially generating unwanted traffic or consuming switch performance must be disabled

at all switches otherwise it would not be possible to unambiguously identify the test packets. The unwanted traffic additionally cause queue filling which influences and distorts the measured data. This unwanted traffic is primarily generated by the services running spanning tree protocols, neighbor discovery protocols, time synchronization, etc. It is also necessary to set up static MAC address entries at both pinging sides to suppress propagation of Address Resolution Protocol (ARP) packets.

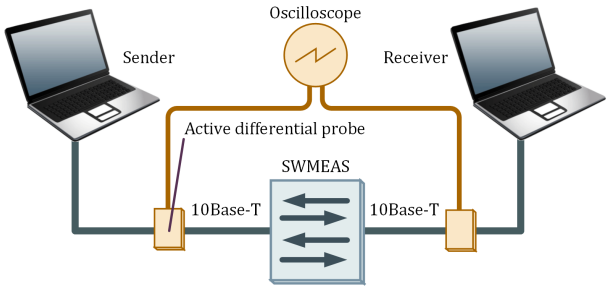


Figure 3.2: Schematic for the first measuring scenario of 10Base-T with the active differential probes.

The wiring differs according to the selected version of the methodology. Figure 3.2 shows the first scenario intended for measurement of the switch fabric latency on Device Under Test (DUT) between 10Base-T Ethernet ports. The time difference is measured between oscilloscope channels, that is connected directly to the transmission medium at L1 via active differential probes. These probes are commercially available, but prototypes for the experimental measurement were used during the evaluation [132]. Where possible, it is necessary to deactivate the Automatic MDI/MDI-X feature, i.e., pair swapping, at the measured ports. This causes considerable difficulties when determining transmitting and receiving pairs. The measurement is usually carried out on *TD+* and *TD-* pairs before and after the switch, in a sender-to-receiver direction.

The readings were designed with respect to RFC 2544 [129] in series of increasing frame lengths (64 B, 128 B, 256 B, 512 B, 1024 B, 1280 B, 1500 B). Each measurement series must be repeated at least 20 times to meet a minimal measurement uncertainty. The higher number of repetitions, the lower the statistical error. An oscilloscope often provides an interface for the automatic delay readings. The threshold voltage level for such automated measurement depends on the particular resistance of used probe. The selection of ports suitable for measuring is detailed in RFC 2889 [133].

To be able to measure the switch fabric latency at higher data rates, it is necessary to modify the wiring diagram due to the aforementioned synchronization. Figure 3.3 demonstrates the wiring diagram, where two auxiliary devices *SWAUX1* and *SWAUX2* were added. These switches must support 10Base-T Ethernet on ingress and egress ports, as indicated in the figure. The arrangement does not directly measure latency between

ingress and engress port of the examined device, but it is still performed directly at L1.

The first step is to measure a delay between directly connected auxiliary switches, i.e., without the DUT in Figure 3.3 noted as *SWMEAS*, and subsequently to create a correction table. This initial measurement is done following the same procedure as described above for all frame lengths and data rates. It is recommended to make significantly more than 20 readings to reduce the measurement uncertainty in the next step of the methodology. Once the table is compiled, the next step is to insert the DUT between those auxiliary ones and repeat all measurements.

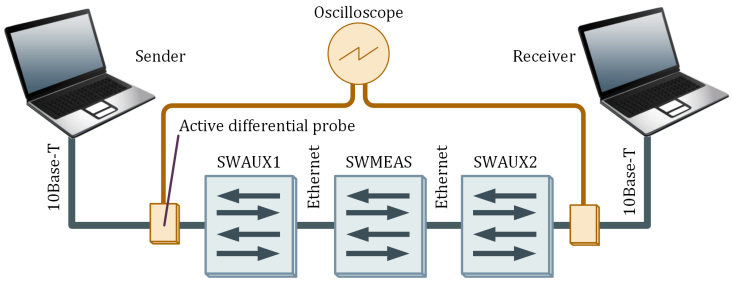


Figure 3.3: Extended schematic for higher data rates. SWAUX 1 and 2 are auxiliary switches and SWMEAS is the examined one.

A mean value of the switch fabric latency is estimated from the set of values obtained by measuring individual series for all frame lengths and data rates. Assuming a normal distribution of the measured data, the latency of 10Base-T Ethernet, is a simple arithmetic mean as written in expression (3.6).

$$\bar{t}_{sw1} = \frac{1}{N} \sum_{i=1}^{N} t_{mes_i} \tag{3.6}$$

where t_{sw1} represents resulting arithmetic mean of the switch fabric latency, N stands for number of measurements in series, t_{mes} are the values obtained from a single measurement.

At the measurement scenario, it is necessary to cleanse results using the compiled correction table. While the correction table consists of arithmetic mean delays for all frame lengths and the examined data rates obtained by the pre-measured series between the auxiliary devices, the correction must be extended to include an additional store-and-forward latency and signal propagation delay at the newly created network segment. This new segment is located between the auxiliary switch and the DUT. The additional latency is not part of the pre-measured characteristics, and thus, it must be estimated. Both latencies can be estimated very accurately, as the store-and-forward latency follows exactly (3.1) and the cable propagation delay remains constant. Whereas the store-and-forward latency is significant, the signal propagation delay can be neglected under general

circumstances. It takes only ∼10 ns with 2 meter cable and is usually significantly smaller than the measurement uncertainty.

It is possible to use a net bit rate, referred as data rate, to estimate the frame input buffering delay introduced by the Store-and-forward method. This assumption can be made as the frame is equipped with the preamble, Start Frame Delimiter (SFD) and Cyclic Redundancy Check (CRC) at L2. These frame fields are encoded together with the rest of the frame. They are explicitly mentioned because they are not usually provided to higher layers such as MAC addresses or *EtherType*. The length of all these fields must be taken into account in the correction estimation. The arithmetic mean for a given frame length is computed as shown in expression (3.7).

$$\bar{t}_{sw2} = \frac{1}{N}\sum_{i=1}^{N} t_{mes_i} - \bar{t}_{aux} - \frac{l_{hf} + l_f}{R_{nb}} - t_{sp} \tag{3.7}$$

where t_{aux} is the mean delay of auxiliary switches taken from correction table [s], l_{hf} is the Ethernet header length including preamble, SFD and CRC (208 bits)[3], l_f is length of the test packet payload, R_{nb} is the net bit rate and finally the optional t_{sp} signal propagation delay.

The subsequent part of the measurement methodology is to determine a measurement accuracy. The overall measurement accuracy is given by an expanded standard uncertainty covering both A and B type [130]. The standard A type uncertainty characterizes a dispersion of measured values. For the first measurement scenario, the A type uncertainty shown in (3.8) can be estimated as the experimental standard deviation of the mean expressed as the sample standard deviation of the mean $s(t_{sw})$ divided by square root of number of measurements $\sqrt{N}$. The expression quantifies how well the t_{sw} estimates the expected mean value.

$$u_A(t_{sw}) = \frac{s(t_{sw})}{\sqrt{N}} = \sqrt{\frac{1}{N(N-1)}\sum_{i=1}^{N}(t_{mes_i} - \bar{t}_{sw})^2} \tag{3.8}$$

At the second measurement scenario, the same expression 3.8 is used to determine the experimental standard deviation of the mean. However, the standard A type uncertainty of the measured values must be expanded by the uncertainty of the correction measurements. This combined uncertainty can be evaluated as the sum of squares of the particular uncertainties for scenarios with, and without, the inserted DUT as shown in

[3]Preamble (7B) + SFD (1B) + Destination MAC (6B) + Source MAC + EtherType (2B) + CRC (4B) = 26 B = 208 b

expression (3.9).

$$u_A(t_{sw2c}) = \sqrt{u_A^2(t_{sw2}) + u_A^2(t_{aux})} \tag{3.9}$$

The combined standard measurement uncertainty can be then determined by a well-known expression (3.10), where $u_B(t_{sw})$ corresponds to a standard B type uncertainty primarily caused by the specific measuring instrument characteristics. It includes oscilloscope parameters such as sampling rate, resolution, skew delay, etc. Eventually, it is necessary to multiply the value of the combined uncertainty $u_C(t_{sw})$ by the coverage factor $k_t = 2$ to obtain the expanded uncertainty and achieve 95% confidence level as shown in expression (3.11).

$$u_C(t_{sw}) = \sqrt{u_A^2(t_{swx}) + u_B^2(t_{sw})} \tag{3.10}$$

$$U = k_t u_C(t_{sw}) \tag{3.11}$$

3.2.4 Analysis of experimental measurements

A wide range of switches from different deployment areas was selected for experimental measurements to verify the proposed methodology. Additionally to the first experimental measurements published in [122], the objective was to test the methodology for 10GBase-R Ethernet including OF switches.

The measurements were accomplished using oscilloscope Tektronix DPO4032 with a maximum sampling frequency of 2.5 GS/s satisfying a required minimum of 100 MS/s. The oscilloscope supports an external network connection, thus, readings were automated via a program written in python language exploiting a PyVISA library [134].

The automated approach allows to obtain measured values remotely from the oscilloscope. Since the latency reading is done automatically by the oscilloscope using the maximum resolution, there is a positive impact on the B type uncertainty. While the lowest measured switch fabric latency was about one microsecond the measurement resolution was set up in nanoseconds. The B type uncertainty was estimated to ∼60 ns for experimental measurements.

Unfortunately, the advanced reading was introduced later to the experimental measurement for switches supporting 10Base-R and OF. This is the reason why readings made manually show a higher expanded uncertainty of measurements as defined in (3.11). Moreover, the standard A type uncertainty also decreased since the process automation enables to take more readings within the same time range.

Thousands of readings for dozens of switch-data rate combinations were taken. All

measurements were made unidirectionally between random ports or ports supporting desired data rate. This specific procedure was selected because randomly performed measurements indicated that the direction of measurement or selected port pairs do not have significant influence on nominal values.

In most cases, the achieved expanded uncertainty for automated measurements is up to 8 % of the estimated mean value. This is primarily due to acquiring more precise readings and increasing the number of readings up to 50 samples. For manual measurements, the achieved expanded uncertainty mostly fluctuates between 10 % and 15 %. If the estimated mean value of the switch fabric latency reaches 1 μs the expanded uncertainty relative to given mean reaches up to 30 % in peak. This is caused by the enlargement of the sampling window, especially for large frames at 100Base-T in the second measurement scenario, since the inserted network segment adds a significant buffering delay.

The correction characteristic of the delay between auxiliary switches had a linear progression at all Ethernet variants as shown in Table 3.1. To estimate correction characteristics, the linear regression was employed. The linearity is confirmed by the value of coefficient of determination R^2 which adheres to 1 for all data rates.

Table 3.1: Correction functions estimated by linear regression.

Ethernet	Correction function	R^2 [-]
10Base-T	y=7.995E-7x+3.344E-5	1.0000
100Base-TX	y=7.978E-8x+1.667E-5	1.0000
1000Base-T	y=7.803E-9x+1.555E-5	0.9996
10GBase-R	y=9.024E-10x+2.402E-5	0.9970

Industrial switches

As industrial Ethernet is one of the most demanding sectors, especially in the field of real-time control, we start with industrial switches. Switches from four vendors were tested. Table 3.2 summarizes results from experimental measurements. It encompasses measured values for selected set of frame sizes. The table compares only 1000Base-T and 100Base-TX Ethernet because 10Base-T Ethernet is now deprecated. However, based on the previous experimental measurements, the observed switch fabric latency for 10Base-T fluctuates around 10 μs with a low expanded uncertainty close to 0.1 μs. The relatively low value of the expanded uncertainty is due to the approach at selected in the first scenario that is truly direct and does not cumulate additional standard A type uncertainty as in the case of the second scenario.

The results show that almost all switches target at the low-latency except Hirschmann RS2 FX/FX which behaves more like L2 software bridge. Values between 2 and 3 μs for 100Base-TX Ethernet are above the average compared to common office switches

Table 3.2: Switch fabric latencies of industrial switches.

Switch	Mode	Frame length [B] 64	256	512	1024	1500
		Switching latency ± U [μs]				
EtherWAN EX83242	1Gbps	0.89 0.25	0.69 0.22	0.80 0.22	1.01 0.23	0.89 0.22
RuggedCom RS900	100Mbps	2.84 0.28	2.78 0.26	2.53 0.33	2.41 0.32	1.93 0.28
RuggedCom RSG2100		2.75 0.36	2.97 0.29	2.36 0.38	2.54 0.40	1.69 0.31
RuggedCom RS930L		2.61 0.27	2.68 0.26	2.53 0.32	2.39 0.32	2.06 0.26
EtherWAN EX83242		1.78 0.21	1.82 0.27	1.49 0.27	1.67 0.27	1.24 0.27
Hirsch. MS20/MM23		5.85 0.38	5.99 0.35	5.59 0.39	4.77 0.35	5.04 0.36
Hirsch. RS2 FX/FX		7.97 0.34	10.70 0.34	15.36 0.37	25.90 0.41	35.28 0.31
Westermo DDW-226		3.31 0.33	3.31 0.27	3.16 0.29	3.36 0.31	2.84 0.35
Westermo Lynx DSS		3.09 0.28	3.28 0.25	2.98 0.31	3.34 0.26	2.83 0.32

presented below. These values point to the optimized internal switching architecture regardless of the additional switch features. Due to the manual readings, the expanded uncertainty reaches up to 0.3 μs, which is up to 20 % of estimated latencies in some cases.

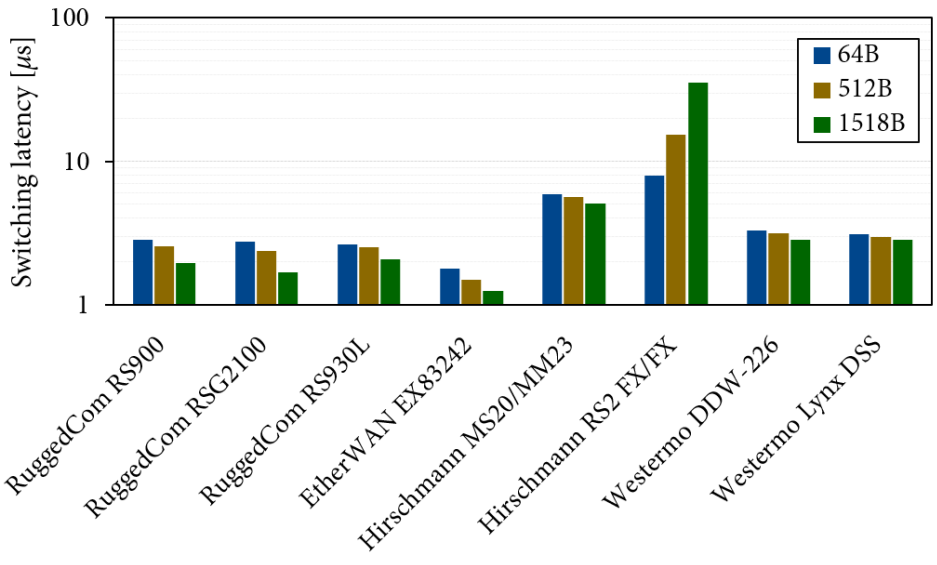


Figure 3.4: Switching latency comparison of industrial switches for 100Base-TX.

The results for 100Base-TX Ethernet is visualized in Figure 3.4. Two observations clearly stand out of the bar chart. The switch fabric latency is gently descending with the increasing frame length. On the contrary, the switch fabric latency at Hirschmann RS2 FX/FX indicates a clear linear increase. Such behavior is inappropriate in the real-time

networking.

The only switch that has proved to reach sub-microsecond is the EtherWAN at 1 Gbps. Even if the expanded uncertainty remains on the border of 0.2 μs it reaches up to 30 % of the estimated switch fabric latency. The bigger part of this uncertainty still reside in the standard A type uncertainty due to the dispersion of readings.

Office switches

The second category comprises a selection of office switches commonly deployed in access-level of campus and enterprise networks. The 10Base-T Ethernet was omitted for the same reasons as industrial switches. Table 3.3 provides an overview of the mean switch fabric latency values for 1000Base-T. The last item, Mikrotik RB2011, was intentionally configured to run L2 software bridge between measured ports. One can clearly observe the rapid linear growth of switch fabric latency similar to the aforementioned Hirschmann switch. This behavior is affected mainly by a high CPU load caused by switching tasks. When comparing the values for 100Base-TX and 1000Base-T, the former yields lower latencies on same ports. While the expanded uncertainty ranges from 0.2 μs to 0.3 μs in the bottom half of the table, the values in the upper half are almost 50% lower. This is due to the fact that these switches were measured already automatically increasing the number of readings to 50.

Table 3.3: Switch fabric latencies of office switches 1000Base-T.

Switch	**Frame length [B]**				
	64	**256**	**512**	**1024**	**1500**
	Switching latency ± U [μs]				
Dell S4810	2.03 0.14	1.97 0.14	2.03 0.14	2.14 0.13	2.01 0.13
Dell 5524	2.04 0.14	2.05 0.14	2.04 0.14	2.18 0.13	2.08 0.13
HP 5406zl	2.80 0.14	3.09 0.13	3.36 0.13	3.74 0.14	4.08 0.12
HP E3800	2.73 0.26	2.76 0.22	3.38 0.21	3.42 0.27	3.67 0.21
3Com 5500-EI	3.55 0.25	2.66 0.21	2.88 0.23	2.86 0.26	2.74 0.23
RB2011LS-IN	1.71 0.25	1.70 0.19	1.66 0.22	1.64 0.26	1.56 0.23
RB2011LS-IN Bridge	16.29 0.36	20.02 0.32	25.65 0.34	34.86 0.41	44.71 0.40

The switch fabric latency from 2 to 4 μs is satisfactory and it can boldly compete with the industrial switches. There are two exceptions both represented by RouterBoards from Mikrotik. While the highly growing latency in case of the bridge was described, the other extreme is the switch fabric latency falling below 2 μs during the common L2 switching.

In this case the expanded uncertainty reaches 16 % of the estimated switch fabric latency. The same conclusions can be made for results in Table 3.4 fot 100Base-TX Ethernet. The only difference is that switch fabric latencies are more spread among particular switches.

Table 3.4: Switch fabric latencies of office switches 100Base-TX.

Switch	**Frame length [B]**				
	64	**256**	**512**	**1024**	**1500**
	Switching latency ± U [μs]				
Dell 5524	4.04 0.13	3.95 0.15	4.14 0.13	4.05 0.13	4.07 0.14
Zyxel GS-105B	2.62 0.30	2.87 0.28	2.30 0.31	2.39 0.27	2.44 0.30
HP 5406zl	3.75 0.25	3.95 0.25	4.03 0.29	4.67 0.24	4.53 0.24
HP E3800	4.01 0.25	4.41 0.25	4.53 0.29	4.92 0.32	4.53 0.28
3Com 5500-EI	6.85 0.59	3.49 0.29	3.15 0.33	2.72 0.36	2.62 0.30
NETGEAR FS105	7.51 0.74	7.18 0.59	7.04 0.61	6.92 0.60	5.53 0.73
Cisco 3500XL	11.93 0.34	12.12 0.32	11.90 0.35	9.98 0.35	10.06 0.30
RB2011LS-IN	2.21 0.26	2.47 0.26	2.16 0.26	2.47 0.29	1.97 0.29
RB2011LS-IN Bridge	15.72 0.45	19.56 0.43	24.30 0.45	35.23 0.59	43.46 0.48

Enterprise switches

Although switches in this category may overlap with the the office category, switches supporting 10GBase-R Ethernet on least up-link ports were marked as enterprise switches for the sake of categorization. In our case, this includes switches with Small Form-factor Pluggable (SFP+) transceiver or with an older version of XFP transceiver. The Dell 5524 was used as the auxiliary switch for all 10 Gbps measurements. It was split into two VLANs forming two auxiliary switches as described in the second measurement scenario. Table 3.5 provides an overview of all results for Dell, HP, Cisco and Foundry vendors.

The measurement results show a high stability of the expanded uncertainty for given switches with 50 readings. The uncertainty is slightly above 0.1 μs in all cases which means up to 6 % relative to the estimated latency. The only exception is Dell S4810 where the switch fabric latency falls down below 1 μs and the expanded uncertainty reaches up to 15 %. In principle, the absolute values can be affected by SFP+ transceivers or by the fact that only up-link ports were available on the tested switches. The measured latencies are visualized in Figure 3.5. The results indicate possible differences in the switch architecture. While most latencies indicate slow linear increase, the latency for Dell switches remains almost constant. This behavior suggest that a shared memory

Table 3.5: Switch fabric latencies of 10GBase-R switches.

Switch	Frame length [B] 64	256	512	1024	1500
	Switching latency ± U [μs]				
Dell 5524	2.05 0.11	2.21 0.11	2.18 0.11	2.07 0.11	2.14 0.12
Dell S4810	0.82 0.12	0.88 0.11	0.94 0.12	0.88 0.12	0.85 0.13
Cisco Catalyst 3750x	4.27 0.11	4.73 0.10	4.91 0.12	5.49 0.12	5.94 0.13
Foundry Edgelron 8x10G	3.27 0.12	3.74 0.11	4.06 0.12	4.76 0.11	5.51 0.11
HP 5406zl	2.07 0.12	2.20 0.11	2.37 0.11	2.85 0.12	3.25 0.11
HP 3800E	1.97 0.11	2.17 0.11	2.28 0.12	2.52 0.11	2.87 0.12

block for both input and output port queues is likely used and that there is no additional frame transmission between line cards. The remaining graphs demonstrate an opposite trend. One can clearly see that increasing frame size also increases switch fabric latency.

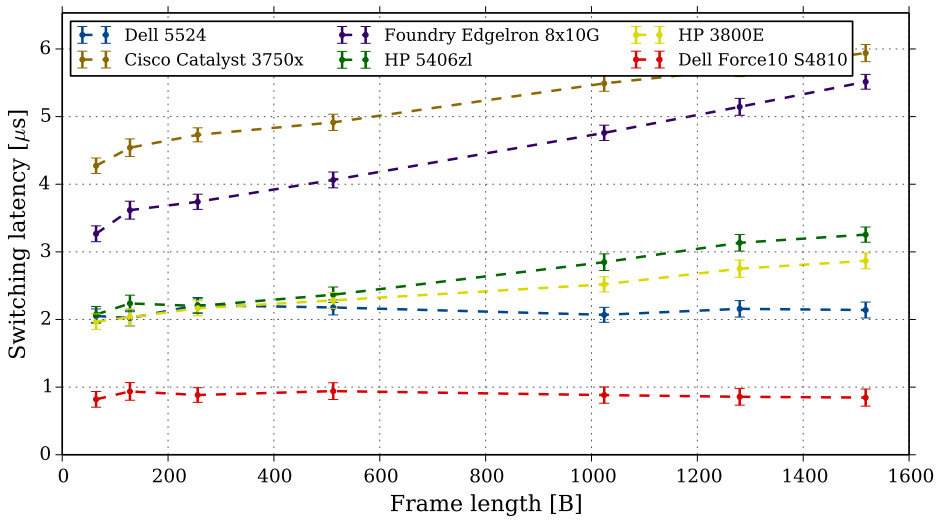


Figure 3.5: Switch fabric latency dependent on the frame length for 10GBase-R.

The absolute values for particular switches are surprisingly high in comparison to lower data rates. This indicates a convergence toward the real switch fabric latency. The phenomenon is illustrated in Figure 3.6, where three switches are placed supporting data rates from 10 Mbps up to 10 Gbps.

OpenFlow switches

In the last part, the measurements were focused on OF switches. As detailed in 2.3, OF is the part of lower layer of the SDN architecture and represents an interface between a

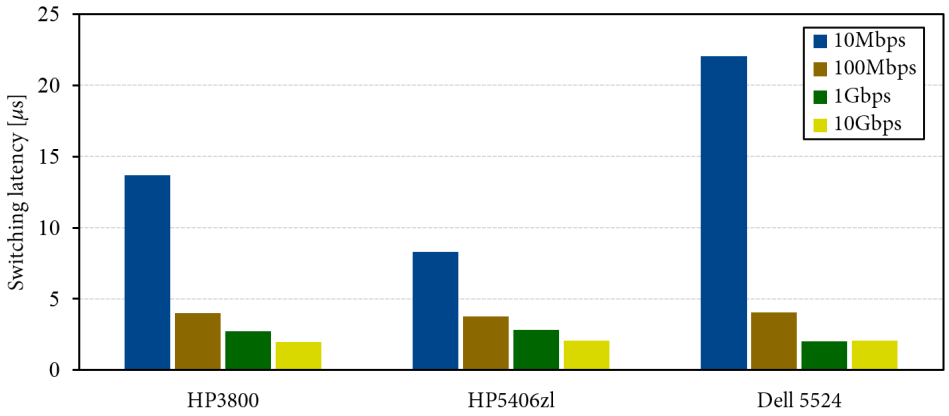


Figure 3.6: Dependency of switch fabric latency on data rate for 64B frames.

logically centralized controller and controlled switches.

Matching rules for the measurement were created just for a destination MAC address as is common with L2 switches. Since the ARP is eliminated by static entries on both client sides it is necessary to upload only two matching rules to DUTs. All other tuples are wildcarded. To perform the measurement, Floodlight controller was chosen with a tool named Static Flow Entry Pusher [135]. The tool is built as a controller module and its interface is accessible via JavaScript Object Notation (JSON) and the controller's web interface. Such approach enables us to setup a time-unlimited matching rule in both directions for test frames. All forwarding modules in the controller were deactivated to prevent any unwanted matching rules to be generated.

Four hardware switches supporting OF and one bare-metal server running an OF service were evaluated. While three switches from HP and Dell had truly integrated OF support in the firmware, the fourth RouterBoard had the OF support in form of the additional software package. The last examined switch was Open vSwitch running on server with dual-core Atom CPU at 1.4 Ghz with 2GB RAM. Debian Wheezy was used as na operating system. The server had two integrated network interface cards up to 1000Base-T Ethernet. The Open vSwitch is a complex SDN switch designed for network virtualuzation and it maintains its own specific flow rule database.

Table 3.6 lists an overview of the results. The estimated mean switch fabric latency on all switches is considerably higher than that on common L2 switches with one exception. These high values are produced by the frame processing and evaluation of matching rules via software, thus, by CPU. On the other hand, not all values are high. The ones fed in by RouterBoard are very close to the values in the bridge mode. It is expected that the switch fabric latency will growth significantly in case of higher switch load, since the CPU time must be shared among other ports.

Even if the Open vSwitch creates its own flow rules derived from OF matching rules

and applies them for a particular traffic, great disproportion is shown between different data rates. This can be caused either by non-optimized Network Interface Card (NIC) drivers or the internal process scheduler. In case of HP switches, it is apparent that the port data rate has no significant effect on the overall switch fabric latency. Unfortunately, we were not able to strictly redirect matching rule processing to hardware in the HP boxes. Moreover, in case of HP 5406zl a new firmware, that unified OF configuration at HP boxes, added up to 100 μs in some cases.

Table 3.6: Switch fabric latencies of OpenFlow switches.

Switch	**Mode**	**Frame length [B]**				
		64	**256**	**512**	**1024**	**1500**
		Switching latency ± U [μs]				
Dell S4810	10Gbps	1.00 0.12	0.98 0.11	0.92 0.11	0.76 0.13	0.76 0.12
HP5406zl		261.90 3.38	244.66 5.17	272.41 8.23	292.00 7.07	272.31 6.94
HP3800E		169.08 1.51	182.43 3.48	170.37 3.57	188.56 4.75	207.00 6.29
Dell S4810	1Gbps	2.18 0.15	1.89 0.13	2.04 0.14	1.98 0.13	2.03 0.13
Open vSwitch		33.09 1.45	36.85 1.22	37.39 1.27	41.77 0.98	46.69 1.01
RB2011LS-IN		15.69 0.59	20.06 0.21	25.22 0.78	35.09 0.21	46.77 2.08
HP5406zl		271.35 5.92	249.70 4.73	277.98 7.18	294.30 6.91	299.27 9.58
HP3800E		179.51 6.88	181.05 3.31	170.67 7.21	197.96 6.10	208.02 6.95
Open vSwitch	100Mbps	233.45 6.32	209.93 5.74	163.46 5.84	88.13 3.98	48.24 0.96
RB2011LS-IN		16.57 0.16	20.89 0.53	26.56 1.09	35.55 0.24	45.08 0.61
HP5406zl		262.28 4.08	232.26 10.48	240.73 9.96	281.57 9.75	301.49 7.35
HP3800E		160.45 0.76	165.26 6.72	196.91 3.12	197.28 4.89	213.56 4.91

As shown in Figure 3.7, the only exception among the evaluated switches is Dell which shows latencies oscillating around 2 μs at 1 Gbps and even below 1 μs at 10 Gbps with the expanded uncertainty close to 0.1 μs. One can notice that OF and non-OF switch fabric latencies are nearly identical. The Dell switch is intended for datacenters and is proclaimed as ultra-low-latency. The great latency stability is the consequence of dedicated CAM block to OF process. All OF matching rules are internally processed as an access list and thus probably highly optimized. Although only one switch gives sufficient values, it shows that the OF could be implemented even in demanding low-latency networks.

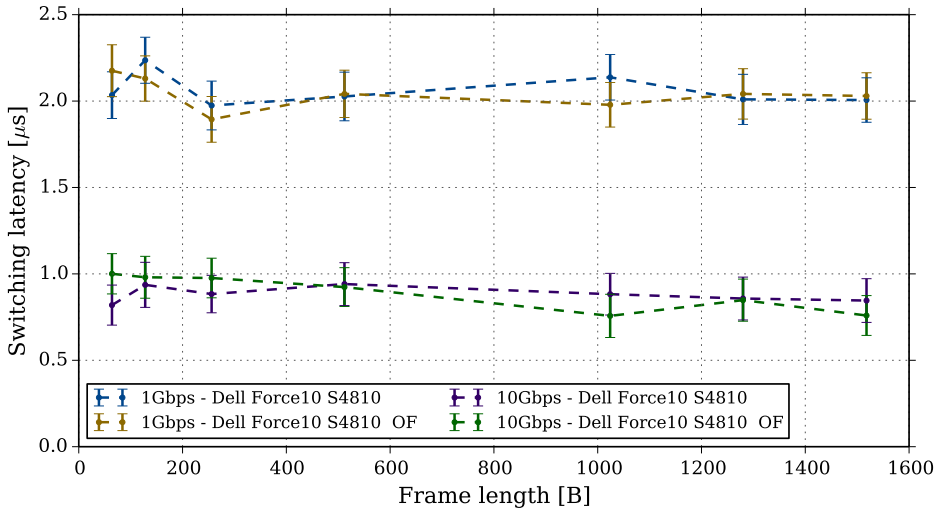


Figure 3.7: Switch fabric latency on Dell S4810 for the OF matches traffic and non-OF switching mode.

3.2.5 Experimental measurement summary

Although many vendors publish switch fabric latencies, the latencies are mostly limited to 64B frames. This may not be precise enough in high demanding industrial networks intended for real-time control. The proposed measurement methodology allows to determine the switch fabric latency by commonly available tools. It proved itself to be usable even for the high data rates as is 10GBase-R with a reasonable expanded uncertainty, which was up to 15 % relative to measured values in the case of automated readings. One can even suggest that this methodology could not only be used for Ethernet but also for other transmission means without significant modifications.

Experimental results across different switch categories can be also advantageously utilized for network simulations. Results obtained for OF-enabled switches suggest feasibility of the SDN concept even in low-latency networks. This conclusion opens the door for SDN not only to industrial networks but also to all low-latency environments based on the Ethernet.

There is always an easy solution to every human problem — neat, plausible, and wrong.

H. L. Mencken

4

LP and LDV Multicast Problem

Although the ultimate goal of this thesis is to formulate an algorithm solving the multi-tree BDLDV multicast problem, this chapter starts with the essential core of the problem. In this first phase, the problem, introduced in Section 1.2, is formalized as an LDV multicast problem.

The LDV multicast problem is classified as centralized, static, and source-based, since multicast group members are known in advance, and data sources produce nearly constant data flow. This classification means that the multicast distribution trees can be precomputed irrespective of the real-time communication requirements. With that condition in mind, it is acceptable to employ LP despite the limited scalability.

Initially, the chapter introduces a concept of mathematical modeling that briefly introduced with a focus on LP problems. After the introduction to the methods used for discrete optimization, a mathematical model of the single-tree LDV multicast problem is proposed and numerically evaluated using random network topologies.

4.1 Mathematical programming

Most of the engineering systems require optimization at some life-cycle point, regardless of whether they are built from scratch or extending the already existing structure. The optimization process seeks to provide the best solution from all feasible solutions to the

problem. Typical real-world tasks are related to the transportation of some medium, e.g., power demand optimization in the grid or minimizing the cost of routing in data networks under various constraints.

As this thesis shows, real-world problems are frequently complex, and as such, there is a need for a more general framework to make them computationally solvable. Mathematical programming models are a perfect tool for providing a robust framework describing the problem formally.

Generally, the optimization target is expressed using a mathematical function known as the *objective function* that depends on *decision variables* whose optimal values are sought. There are restrictions to be satisfied denominated as the *feasible region* that is formally defined by *constraints* of the problem [136].

As indicated above, the mathematical model encompasses three components [137]:

Decision variables are variables within the model that can be controlled, usually denoted $x_1, x_2, ..., x_n$.

Objective function is the function that we want to optimize (maximize or minimize) denoted $z(x_1, x_2, ..., x_n)$.

Constraints are conditions or limitations of the problem. Every condition is expressed through equality or inequality, e.q., $S(x_1, x_2, ..., x_n) \leq 0$.

The formal structure of the mathematical model is expressed in (4.1).

$$\begin{aligned} &\textit{minimize} && \text{objective function} \\ &\textit{subject to} && \text{constraints} \end{aligned} \tag{4.1}$$

A taxonomy of mathematical programming problems differs depending upon the type of variables and the nature of the objective function. If the variables are continuous and both the objective function and constraints are linear, the problem is termed as an LP problem. If any variable is of integer or binary type, while both the objective and constraints are linear, the problem is Mixed-Integer Linear Programming (MILP). In case all variables are of integer type, the problem is reduced to ILP. Non-linear problems keep the same taxonomy.

The indisputable advantage of mathematical programming is the clarity of the problem description allowing models of complex engineering systems to be generalized. Moreover, the model may involve both operation and planning in one place. Such a model can be evaluated iteratively as the engineering system evolves and optimize it under changing conditions. A potential disadvantage is the limited scalability of such mathematical

programming models, reducing the usability in real-time tasks. The way to tackle the scalability issue is a topic for the following chapters.

4.1.1 Linear programming problem

G. B. Dantzig formulated the LP problem around 1947 during his work for the United States Air Force when developing a mechanized planning tool for a time-staged deployment, training, and logistical supply [138]. Although he was overtaken by L. V. Kantorovich dealing with a similar type of problem in 1939, the origin of so-called *Programming in Linear Structure* is credited to G. B. Dantzig.

A linear programming problem occurs when both the objective function and constraints are linear, and variables are continuous. Any given LP problem can be put in different forms applying appropriate manipulations. Preferable forms are the standard format and canonical format. The standard format occurs if all restrictions are equalities and all variables are non-negative. The canonical format occurs if all variables are non-negative and all the constraints are inequalities ($\leq$ for maximization, $\geq$ for minimization) [138]. As some of the methods used in LP requires the standard format, e.g., a simplex method detailed in the following text, a transformation from canonical to standard format is possible using slack variables. The slack variable is a non-negative variable added to an inequality constraint that is then converted to equality.

The general canonical form of the minimization LP problem is defined as shown in equations (4.2)–(4.6). The equation in (4.2) describes the objective function where $x_1, x_2, ..., x_n$ are decision variables, and $c_1, c_2, ..., c_n$ are known constants, e.g., cost coefficients. Then constraints in (4.3)–(4.3) (e.q. functional, structural or technological restrictions) then follow, where coefficients a_{ij} for $i = 1, ..., m$, $j = 1, ..., n$ are called technological coefficients [138]. The definition is closed by a set of special constraints in (4.6), termed as sign restrictions, dealing only with decision variables.

$$\begin{aligned}
&\textit{minimize} && c_1x_1 + c_2x_2 + \ldots + c_nx_n && && (4.2)\\
&\textit{subject to} && a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n && \geq b_1 && (4.3)\\
& && a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n && \geq b_2 && (4.4)\\
& && \vdots && \vdots &&\\
& && a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n && \geq b_m && (4.5)\\
& && x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0 && && (4.6)
\end{aligned}$$

Analogously, the minimization LP problem can also be formulated by a matrix ex-

pression as in equations (4.7)–(4.9).

$$\begin{aligned} &\textit{minimize} && c^T x && (4.7)\\ &\textit{subject to} && \mathbf{Ax} \geq \mathbf{b} && (4.8)\\ & && \mathbf{x} \geq 0 && (4.9) \end{aligned}$$

where

$$\begin{aligned} \mathbf{x^T} &= [x_1, \cdots, x_n] && (4.10)\\ \mathbf{b^T} &= [b_1, \cdots, b_m] && (4.11)\\ \mathbf{c^T} &= [c_1, \cdots, c_n] && (4.12)\\ \mathbf{A} &= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} && (4.13) \end{aligned}$$

Basic LP problem

To give a better sense of the formal LP description, we define a simple LP problem. The optimization problem aims to maximize the objective function in (4.14) under several constraints defined in (4.15)–(4.19). Using this trivial example, we can demonstrate a geometry representation of a two-dimensional problem and employ a geometrical approach to find the best solution.

$$\begin{aligned} &\textit{maximize} && x + y && (4.14)\\ &\textit{subject to} && 3x + 2y \leq 11 && (4.15)\\ & && -2x + 9y \leq 18 && (4.16)\\ & && -x - 2y \leq -3 && (4.17)\\ & && x \geq 0 && (4.18)\\ & && y \geq 0 && (4.19) \end{aligned}$$

The defined LP problem consists of two decision variables, i.e., it is two-dimensional. Problem constraints form halfplanes that split the solution space. As one can see in Figure 4.1, the intersection of halfplanes, a polygon, determines the feasible region of

the problem. The constraints are boundaries demarcating the feasible region and at each intersection of these boundaries is a corner point. Generally, if the problem has an optimum solution and at least one corner point of the feasible region exists, an optimum solution is one of the corner points [137].

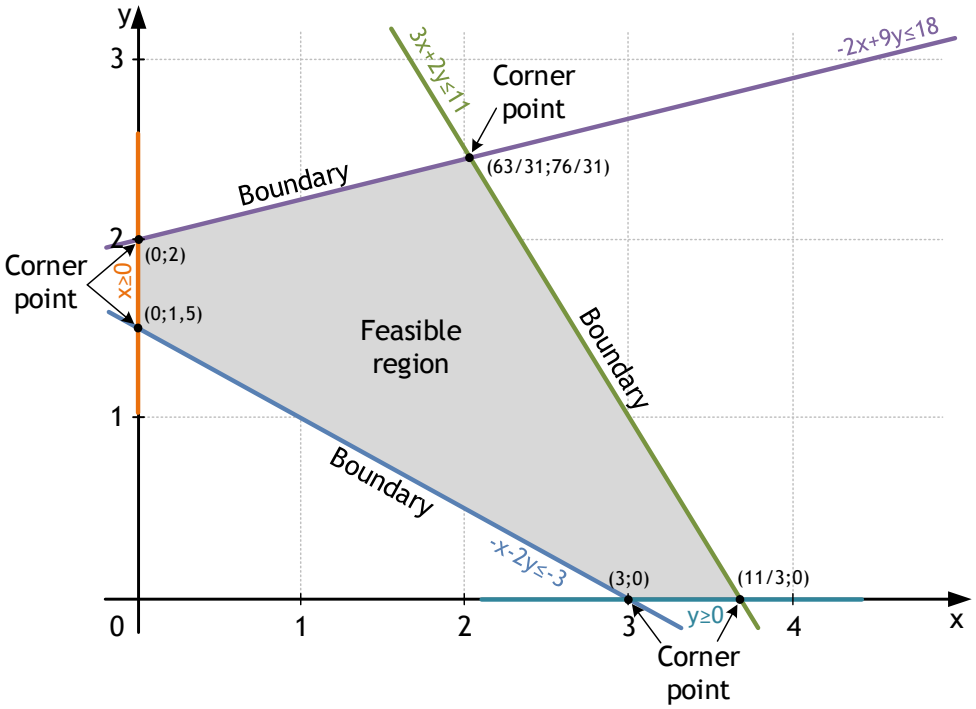


Figure 4.1: Geometrical representation of the example problem defined in (4.14)–(4.19), and nomenclature used in LP.

Figure 4.2 illustrates the geometrical approach to find the best solution to the LP problem. If we move the objective function $z = x + y$ up along the y axis while keeping the slope, z increases (maximization). Moving the objective function, we visit step by step all corner points where z acquires values $\frac{3}{2}$, 2, 3, $\frac{11}{3}$, and $\frac{139}{31}$. The solution is obvious from the list: $\max_{x,y} z(x, y) = \frac{139}{31}$.

The above claims remain valid even for n-dimensional LP problems. In the case of three-dimensional space, the feasible region in the form of a polyhedron is bordered by facets given by the intersection of halfspaces defined by problem constraints. An objective function is a plane whose optimum lies in the intersection of the plane with some of the polyhedron's corner points. In the general case of n-dimensional space, the objective function is expressed by a hyperplane and the feasible region is a polytope[1] bordered by hyperplanes associated with given constraints.

[1]A d-polytope is a geometric object that a) is the convex hull of finitely many points in $\mathbb{R}^d$, or b) is the bounded intersection of finitely many half-spaces in $\mathbb{R}^d$.

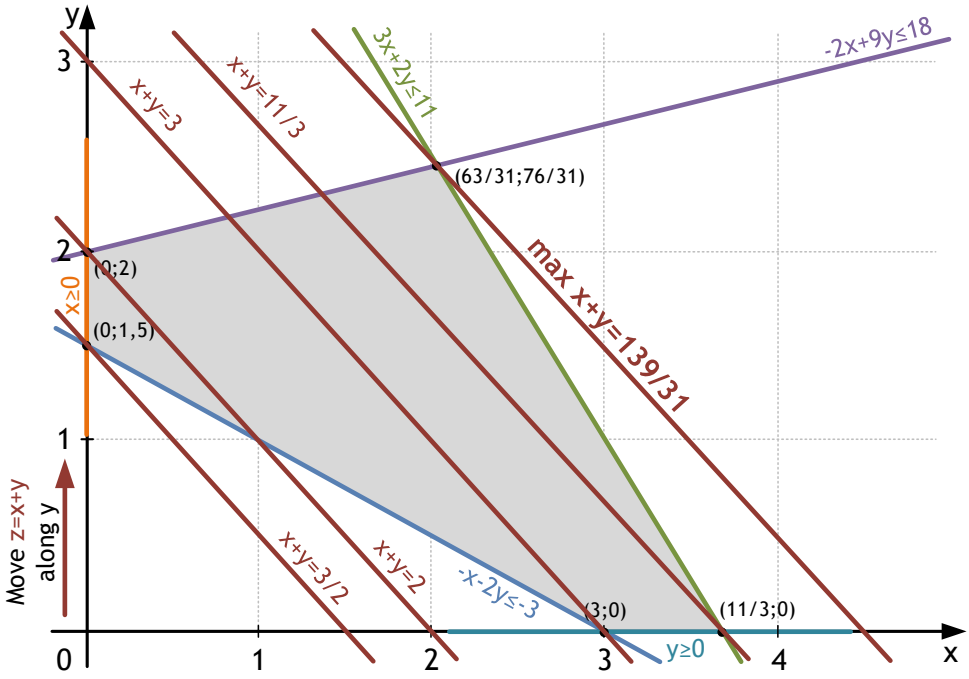


Figure 4.2: Geometrical approach to solve an LP problem. The maximized objective function $z = x + y$ is moved to the corner points where the point of intersection at $(\frac{63}{31}; \frac{76}{31})$ is the problem maximum $z = \frac{139}{31}$.

Simplex method

It is evident that a high number of variables and constraints rapidly increase the problem's complexity, as obtaining all corner points and corresponding objective values is significantly demanding. To overcome the limitation given by the enormous number of corner points, G. B. Dantzig invented the *simplex method* which efficiently explores the solution space of the LP problem. The method was firstly published in [139], and it is incorporated as an essential method in most solvers nowadays.

The core idea of the simplex method is built on the fact that the optimum solution is at least one of the corner points. The simplex method is an iterative process in which we start with a solution that satisfies the equations and non-negativities in the standard format of the LP problem and then look for a new solution which is better, in the sense that it has a larger (minimization problem) objective function value. This process continues until it arrives at a solution that cannot be improved. This final solution is then an optimal solution [140]. The key to the simplex method lies in recognizing the optimality of a given corner point solution based on local considerations without having to (globally) enumerate all corner points or basic feasible solutions [138].

Using the simple LP problem defined in (4.14)–(4.19), we can demonstrate the simplex method. Starting in the corner point $(0; 1, 5)$, we choose both neighboring corner points as these increase the value of the objective function (maximization problem). If we choose the point $(0; 2)$, the only option is then $(\frac{63}{31}; \frac{76}{31})$. As the next corner point does not

increase the value of the objective function the algorithm stops. Choosing as the next corner point $(3;0)$ after the starting point, the path is as follows $(0;1,5) \rightarrow (3;0) \rightarrow (\frac{11}{3};0) \rightarrow (\frac{63}{31};\frac{76}{31})$. As the example shows, using both paths, the simplex method ends in the optimum solution.

Although the simplex method effectively finds the best solution, it applies only to LP problems. Considering the integer nature of an ILP or MILP problem, the presented methods are not usable as the optimum solution may lie anywhere in the feasible region. The number of solutions is countable, but the number of points to consider can be enormous, thus inspecting all possible solutions is ineffective. To overcome this issue, the linear relaxation[2] of the integer variables is used and then the simplex method is applied to the resulting LP problem. Usually, this step is a subroutine of some algorithm taking care of the solution space exploration, as a *branch-and-bound* algorithm.

4.1.2 LP-network problems

The optimization of network flow problems may be very challenging as it requires a combination of several skills. At first, there is graph theory that describes the underlying network. Then, the mathematical modeling required to transfer a real-world problem into a formalized mathematical framework and having the model we can proceed to the LP and apply it to solve the model and find the best solution.

From the perspective of network flow problems, the essential is the *minimum-cost flow problem* which can be viewed as a superset of other network flow related problems. As summarized in [137], the *minimum-cost flow problem* aims to find the traffic flows whose cost is minimized to satisfy a traffic demand from a source node to a destination node, under the constraint that the traffic volume passing each link cannot exceed the link capacity.

There are two modifications to the problem that can be derived. Removing the capacity constraint, the *minimum-cost flow problem* is reduced to the well-known *shortest path problem*, efficiently solvable by Dijkstra's algorithm [141]. The second modification is when all costs are set equal to zero; then, the problem is reduced to a *maximum flow problem*[3].

The solution of the *maximum flow problem* is a solution to the *minimum cut problem* and vice versa. The cut is a partition of a graph into two disjoint sub-graphs. The

[2]The linear relaxation, or linear programming relaxation, is a linear programming problem obtained by dropping the integrality constraint [140], e.g., $x_i \in \{0;1\}$ is relaxed to $0 \leq x_i \leq 1$.

[3]Back in the 1950s of the 20th century, the Soviet railway system roused the interest of the US Air Force. T.E. Harris, in conjunction with General F.S. Ross, formulated a simplified model of railway traffic. This model was a motivation for Ford and Fulkerson to work on the *maximum flow problem*. However, the authors of the problem were more interested in the *minimum cut problem*, i.e., interdiction of the Soviet railway system [142]

maximum flow of a network is limited by the minimum capacity of the cut, the bottleneck of the network. Such information may be valuable not only for optimization during network planning but for a potential attack on the network as well, as it can affect the network with the highest impact. The *maximum flow problem* is solvable using Ford and Fulkerson's original labeling method [143]. However, specific primal simplex implementations have been found to perform more efficiently in terms of speed up and memory usage [138].

Minimum-cost flow problem

To clarify the approach to LP-network problems, we detail the minimum-cost problem which is a steppingstone for more complex problems introduced later in this thesis.

At first, we need to define the underlying network model as a graph. The network is represented by a directed connected graph $G = (V, L)$ where V is a set of network nodes (vertices), and L is a set of network links (edges). A link from node u to node v is denoted $(u, v) \in L$. Each link (u, v) is associated with a transport capacity c_{uv}, and transport cost d_{uv}. The forwarded bandwidth (traffic volume) x_{uv}^{sd}, decision variable, from source node $u \in V$ to destination node $v \in V$ and routed via $(u, v) \in L$ is limited by link capacities $0 \leq x_{uv}^{sd} \leq c_{uv}$. Then, the minimum-cost flow LP problem defined in (4.20)–(4.23) minimizes the cost required to forward bandwidth b from source node $s \in V$ to destination node $d \in V$.

$$\textit{minimize} \quad \sum_{(u,v)\in L} d_{uv} x_{uv} \tag{4.20}$$

$$\textit{subject to} \quad \sum_{v:(u,v)\in L} x_{uv} - \sum_{v:(v,u)\in L} x_{vu} = b \qquad \text{if } u = s \tag{4.21}$$

$$\sum_{v:(u,v)\in L} x_{uv} - \sum_{v:(v,u)\in L} x_{vu} = 0 \qquad \forall u \neq s, d \in V \tag{4.22}$$

$$0 \leq x_{uv} \leq c_{uv} \qquad \forall (u, v) \in L \tag{4.23}$$

The meaning of the objective function in (4.20) is self-explanatory. The objective function minimizes the cost of flow bandwidth from node $s \in V$ to node $d \in V$ in the network. The constraint defined in (4.23) limits the capacity usable by decision variables, as indicated in the previous text. The important parts are constraint equalities (4.21) and (4.22), known as *flow conservation*. The first constraint (4.21) expresses that source node $s \in V$ is the only producer of bandwidth b in the network. The second constraint (4.22) defines that any ingress traffic $\sum_{v:(v,u)\in L} x_{vu}$ to a node, where $u \neq s, d$, must leave the node

as egress traffic $\sum_{v:(u,v)\in L} x_{uv}$ to conserve the flow[4]. There is no need to define additional constraint for ingress traffic to destination node $d \in V$, as this is already ensured by the previous constraints.

As the basics of LP were clarified, we may now consider the framework and methods described above to be a tool for solving the LDV multicast problem, and we can proceed to our research.

4.2 Single-tree LDV multicast problem

Concerning already published algorithms presented in Section 2.5.2, the author of this thesis proposes an ILP formulation minimizing the variation of propagation delay along branches of a single multicast tree. The single expresses the fact that no resources, in terms of bandwidth, are shared along with the tree links. In this basic formulation, only one multicast group at the time is considered. In contrast to some papers, network nodes acting as subscribers can forward the traffic to further nodes as well because nodes represent Ethernet switches.

The goal of the following sections, originally published in [144], is to investigate the impact of various random graph topologies on qualitative parameters of computed multicast trees.

4.2.1 Mathematical Formulation

The following text describes a core ILP formulation and network model used later for the BDLDV problem. To compare qualitative parameters on various graphs and setups, we present the ILP formulation for the LDV problem, as well as the ILP formulation of an agnostic approach based on SPT. The delay and delay-variation constraints are not considered in any of the following mathematical formulations yet.

Network Model

Let's consider directed connected graph $G = (V, L)$ where V is a set of network nodes and L is a set of network links. The set of nodes V represents inter-connecting nodes, e.g., Ethernet switches. The publisher and subscribers are connected to these nodes. The multicast tree $T(v_p, S)$ is a sub-graph of G compounded of a multicast source node (publisher) $v_p \in V$, and multicast destination nodes (subscribers) $S \subseteq V \backslash \{v_p\}$ where the set $S \cup \{v_p\}$ is called the multicast group. Set S and publisher node v_p are interconnected by links through a subset of Steiner tree nodes $M \subset V$ which form a part of $T(v_p, S)$.

[4] Also known as nodal balance, Kirchhoff's first law, nodal rule, etc.

All links are bidirectional, each directed link $\ell = (u, v), \ell \in L$ going from $u \in V$ to $v \in V$ has a counterpart $\ell' = (v, u)$ in the opposite direction from $v \in V$ to $u \in V$. Each node $v \in V$ is incident to a set of ingress links $\omega^+(v)$ and egress links $\omega^-(v)$. A real non-negative value is assigned to every link $\ell \in L$ in the form of a link delay $d_\ell \rightarrow R^+$. Link delay function d_ℓ is a measure of link propagation delay. The function is naturally symmetrical, therefore $d_\ell = d_{\ell'}, \ell \in L, \ell' \in L$.

Let $P_T(v_p, s), s \in S$ be a set of links $\ell \in L$ on a path from node v_p to node s in the tree $T(v_p, S)$ and $M(P_T(v_p, s)) \subset V$ is a set of Steiner nodes along this particular path. The total end-to-end transmission delay $D_T(v_p, s)$ is then a sum of all link delays along the path as given in expression (4.24).

$$D_T(v_p, s) = \sum_{\ell \in P_T(v_p, s)} d_\ell \tag{4.24}$$

The delay-variation δ_T of the multicast tree $T(v_p, S)$ is defined as the maximum difference among end-to-end delays along paths of all node pairs in $v_p \times S$ as is described by expression (4.25).

$$\delta_T(v_p, S) = max\{|D_T(v_p, u) - D_T(v_p, v)||\forall u, v \in S\} \tag{4.25}$$

SPT Formulation

At first, we define an ILP formulation producing SPT with a root node in the multicast publisher v_p. The objective is to minimize total tree size, as defined by expression (4.26), where $y_\ell \in \{0, 1\}$ with $y_\ell = 1$ if traffic from v_p to $v_s \in S$ is forwarded on link ℓ.

$$\min \sum_{l \in L} y_l \tag{4.26}$$

The formulation on the approach uses flow constraints as shown in (4.27). Each flow, from the publisher to a subscriber, is a set of node pairs $\mathcal{PS} = \{\{v_p, v_s\}|v_s \in S\}$, and these sets of node pairs are used for the calculation of the objective function. A flow at link ℓ from node v_p to destination $v_s \in S$ is denoted as φ_ℓ^{ps} and this variable can take a value of forwarded bandwidth b, i.e., φ_ℓ^{ps} is defined in positive domain (4.31).

$$\sum_{\ell \in \omega^+(v)} \varphi_\ell^{ps} - \sum_{\ell \in \omega^-(v)} \varphi_\ell^{ps} = \begin{cases} b & \text{if } v = v_p \\ -b & \text{if } v = v_s \\ 0 & \text{otherwise} \end{cases}$$

$$v \in V, v_s \in S \tag{4.27}$$

The rest of the ILP formulation in (4.28)–(4.30) ensures that the found solution will be a tree aggregating all flows through the binary vector y_ℓ.

$$\varphi_\ell^{ps} \leq y_\ell \qquad (p,s) \in \mathcal{PS}, \ell \in L \tag{4.28}$$

$$y_\ell < \varphi_\ell^{ps} + 1 \qquad (p,s) \in \mathcal{PS}, \ell \in L \tag{4.29}$$

$$\sum_{\ell \in \omega^+(v)} y_\ell \leq 1 \qquad v \in V \tag{4.30}$$

$$\varphi_\ell^{ps} \geq 0 \qquad (p,s) \in \mathcal{PS}, \ell \in L \tag{4.31}$$

LDV Formulation

The objective of the LDV multicast problem is to minimize variation of total propagation delay along all paths in $T(v_p, S)$, as expressed in (4.32).

$$\min(\delta_T(v_p, S)) = min(\delta_{T_{max}} - \delta_{T_{min}}) \tag{4.32}$$

The ILP formulation of the LDV multicast problem comes from the SPT formulation, but this model is extended by a set of constraints detailed in this section. The expression (4.33) tightens $\delta_T(v_p, S)$ for all pairs in $\mathcal{PS}$ using link delays defined in vector d_ℓ. In order to map propagation delays to links selected in a solution process, an additional conversion vector x_ℓ^{ps} is applied in formulations (4.34)–(4.35). The vector x_ℓ^{ps} is defined in (4.36), with $x_\ell^{ps} = 1$ if a flow is forwarded at link ℓ from node v_p to $v_s \in S$.

$$\delta_{T_{min}} \leq \sum_{\ell \in L} d_\ell x_\ell^{ps} \leq \delta_{T_{max}} \qquad (p,s) \in \mathcal{PS} \tag{4.33}$$

$$\varphi_\ell^{ps} \leq x_\ell^{ps} \qquad \ell \in L, (p,s) \in \mathcal{PS} \tag{4.34}$$

$$x_\ell^{ps} < \varphi_\ell^{ps} + 1 \qquad \ell \in L, (p,s) \in \mathcal{PS} \tag{4.35}$$

$$x_\ell^{ps} \in \{0,1\} \qquad (p,s) \in \mathcal{PS}, \ell \in L \tag{4.36}$$

The modification of the objective function from (4.26) to (4.32) may cause the emergence of loops in the final solution. Therefore, we introduce auxiliary constraints that help to avoid stand-alone loops in the solution. The primary constraint (4.41) assures that all links assigned to a particular flow $\varphi_{\ell_i}^{ps}$ are virtually labeled in non-decreasing order in vector o_{ℓ}^{ps}. The constraints are limited only to a set of neighboring pairs of ingress/egress links $\mathcal{IO} = \{\{\ell_i, \ell_o\} | \ell_i \in \omega^+(v), \ell_o \in \omega^-(v), v \in V\}$. The constraints (4.37)–(4.39) express that ℓ_i and ℓ_o are ingress and egress links along a specific flow $(p, s) \in \mathcal{PS}$. Typically, this information can be obtained by logical operation AND for these decision variables. However, operation AND is a non-linear operation; therefore, we apply a standard linearization approach. To accomplish this side step, we have used an auxiliary variable a^{ps}, defined in (4.40), that bonds similarly to x_{ℓ}^{ps} links with an assigned flow and the order constraint.

$$\varphi_{\ell_i}^{ps} - b \geq a_i^{ps} \qquad \ell_i \in \omega^+(v), v \in V, (p,s) \in \mathcal{PS} \tag{4.37}$$

$$\varphi_{\ell_o}^{ps} - b \geq a_o^{ps} \qquad \ell_o \in \omega^-(v), v \in V, (p,s) \in \mathcal{PS} \tag{4.38}$$

$$a_i^{ps} + a_o^{ps} - 1 \leq a_{io}^{ps} \quad (i,o) \in \mathcal{IO}, (p,s) \in \mathcal{PS} \tag{4.39}$$

$$a_{io}^{ps} \in \{0,1\} \qquad (i,o) \in \mathcal{IO}, (p,s) \in \mathcal{PS} \tag{4.40}$$

$$o_{\ell_i}^{ps} - o_{\ell_o}^{ps} \geq a_{io}^{ps}$$
$$(i,o) \in \mathcal{IO}, \ell_i \in \omega^+(v), \ell_o \in \omega^-(v), v \in V, (p,s) \in \mathcal{PS} \tag{4.41}$$

4.2.2 Evaluation

The variation of packet propagation delay is impacted by various network parameters such as network topology and its parameters. Four different random models were chosen with respect to the sufficient variability of evaluated instances. Random graph topologies were generated using models detailed in section 2.4.1. Graphs and main robustness characteristics are listed in Table 4.1. All values are means from 10 generated graph instances. The probability of link selection in Erdős-Rènyi is 7 %. Each node in Watts-Strogatz is connected to 3 nearest neighbors in a ring topology and each link is rewired with a probability of 30 %. Each new node in Barabási-Albert is attached by 3 links to existing nodes. Graph characteristics presented in the table describe fundamental properties. Refer to [80] for a detailed explanation.

Each graph topology instance was 10 times randomly generated with similar parameters for the given model, and for each instance was generated 9 groups were generated with uniformly placed subscribers in a range from 10 % to 90 % of total graph nodes.

Table 4.1: Graph models used for evaluation purposes.

Graph model	$\|V\|$	$\|L\|$	Average nodal degree	Diameter	Link Connectivity	Link Density	Links Per Node	Nodal Connectivity
Barabási-Albert	10	42.0	8.40 ± 3.11	2.60 ± 0.52	2.20	0.47	4.20	2.20
	15	72.0	9.60 ± 4.51	3.00 ± 0.38	2.10	0.34	4.80	2.10
	20	102.0	10.20 ± 5.48	3.10 ± 0.32	2.10	0.27	5.10	2.10
Dorogovtsev-Mendes	10	34.0	6.80 ± 3.67	2.80 ± 0.63	2.00	0.38	3.40	2.00
	15	54.0	7.20 ± 3.96	3.80 ± 0.63	2.00	0.26	3.60	2.00
	20	74.0	7.40 ± 5.01	4.20 ± 0.42	2.00	0.19	3.70	2.00
Erdős-Rènyi	10	21.0	4.20 ± 1.98	5.10 ± 0.74	1.10	0.23	2.10	1.10
	15	31.4	4.19 ± 1.97	7.70 ± 1.42	1.00	0.15	2.09	1.00
	20	44.4	4.44 ± 4.44	9.00 ± 1.83	1.00	0.12	2.22	1.00
Watts-Strogatz	10	20.0	4.00 ± 1.21	5.70 ± 0.67	1.10	0.22	2.00	1.10
	15	30.0	4.00 ± 1.46	9.20 ± 1.14	1.00	0.14	2.00	1.00
	20	40.0	4.00 ± 1.46	11.60 ± 1.65	1.00	0.11	2.00	1.00

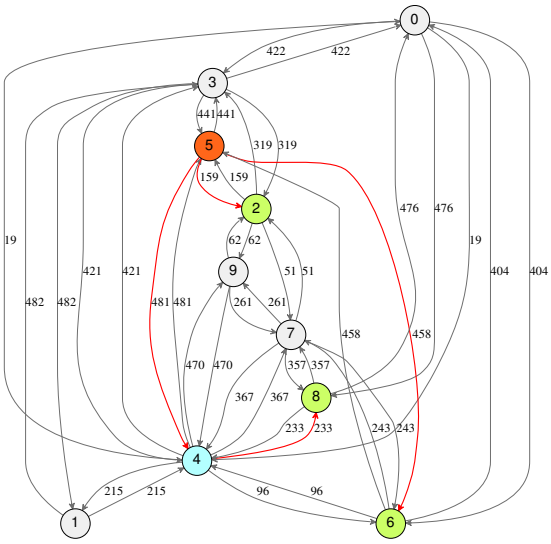
A propagation delay was randomly assigned to each link in a range from 5 to 500 ns. This propagation delay is approximately proportional to a delay on Ethernet segments with lengths from 1 to 100 m. The multicast publisher was randomly placed in a graph center. Due to the exponential time complexity of the ILP, the network size was limited. Depending on the network model and link density, it was possible to numerically evaluate, in a reasonable time, instances with sizes ranging from 10 to 20 nodes.

Numerical results

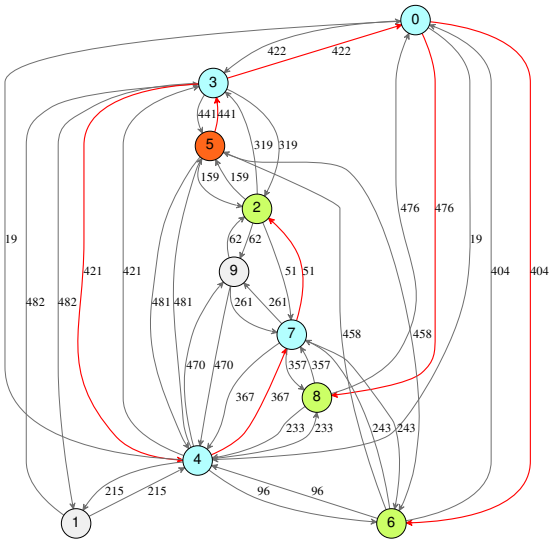
In order to show the benefits of the proposed algorithm, the agnostic-based SPT model and objective-aware LDV model were implemented. Formulations proposed in Section 4.2.1 were written in Open Programming Language (OPL) language and evaluated using the CPLEX Optimizer. Implemented models were run on 21600 instances in total. The outputs from these instances are statistically evaluated and compared.

Results of the LDV model give the best possible solution for a given network and multicast group. The resultant configuration can be proactively deployed into the SDN-enabled network and with proper QoS settings eventually converging to the desired LDV multicast tree. On the other hand, the SPT approach can be seen as the best approximation of an arbitrary SPT protocol. The difference in solutions provided by both models is shown in Figure 4.3.

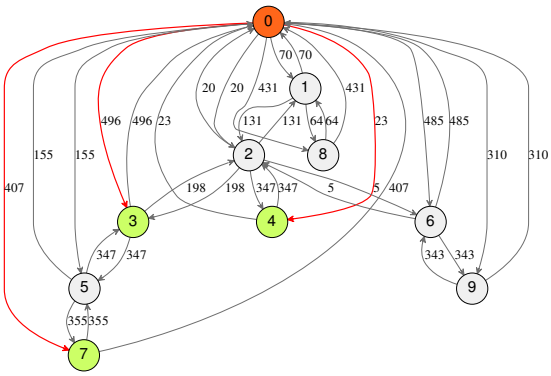
At first, an effect of the multicast group size on the achieved delay variation depicted in Figure 4.4 is analyzed. LDV delivers better results than the SPT approach. The difference can be seen in lower delay variation for LDV, mainly due to the character of graph models. The Watts-Strogatz network has the highest diameter and lowest nodal connectivity, thus, it produces solutions with long paths. Simply, there are not enough alternative paths in the topology. On the other hand, the Barabási-Albert model shows much better results



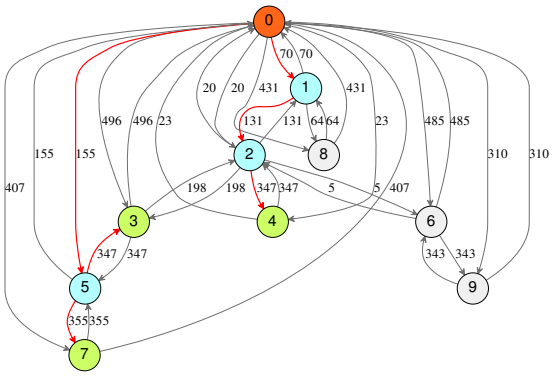
(a) SPT for a Barabási-Albert graph.



(b) LDV multicast tree for a Barabási-Albert graph.



(c) SPT for a Dorogovtsev-Mendes graph.



(d) LDV multicast tree for a Dorogovtsev-Mendes graph.

Figure 4.3: An example of the difference between solutions found by SPT-based 4.3a, 4.3c and LDV formulations 4.3b, 4.3d for Barabási-Albert model and Dorogovtsev-Mendes models with $n = 10$ and 30% penetration of subscribers. In each graph, the orange node is the multicast publisher, green nodes are multicast subscribers, and blue nodes represent Steiner nodes. Numbers next to links represent their weight in ns.

since the model generates a lot of links and the LDV can utilize alternative paths. The remaining two topologies show differences in delay variation somewhere in the middle, proving that the number of links in a graph is a major factor in this setup. The impact of multicast group size is evident. The higher the penetration of subscribers, the greater the delay variation. Growth slows down with the amount of subscribers, as the number of available links decreases.

The second case, where the effect of the multicast group size on a mean path delay $D_T(v_p, u), u \in S$ is investigated, is rather opposite in its progress. The chart in Figure 4.5 shows an unusual drop in the path delay for instances with low penetration of subscribers at graph models with the power-law distribution of node degrees (Barabási-

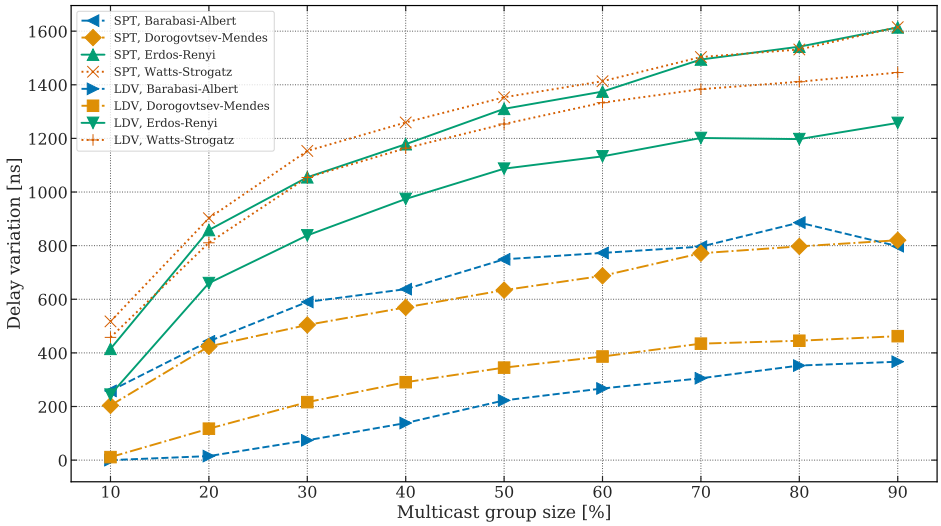


Figure 4.4: Effect of multicast group size on the mean value of least delay variations at graph size $n = 20$.

Albert, Dorogovtsev-Mendes). The path delay decreases significantly at LDV in contrast to theSPT approach. The LDV uses more links, particularly in the beginning when link variability is higher; therefore, it produces paths with higher propagation delays. As the network topologies are always finite, the number of links in the solution is limited as well, and paths cannot grow to infinite lengths. Due to the uniformly placed multicast subscribers, SPT fluctuates almost at constant levels in all graph instances.

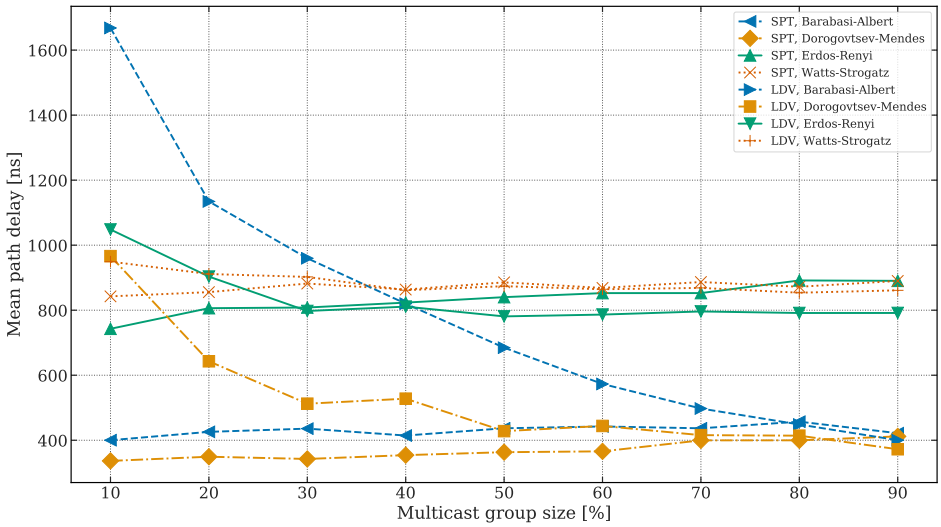


Figure 4.5: Effect of multicast group size on the mean value of path delays at graph size $n = 20$.

Although higher link density gives better results concerning LDV, the multicast tree can contain paths infeasible from a jitter perspective. Since none of the ISO/IEC/IEEE 8802-3 Ethernet QoS mechanisms can guarantee exact priority packet handling (fabric

latency, various queue mechanisms), each node added to the solution potentially increases jitter along the path to a multicast subscriber.

The impact of higher link variability on instances with lower penetration of subscribers is depicted in Figure 4.6. The LDV on the Barabási-Albert model produces trees with a greater number of links, i.e., a higher number of hops a multicast packet has to pass. On the other hand, the chart proves that the SPT formulation produces trees with a lower number of links in all cases. All curves converge at the higher number of subscribers, since it is not possible to build a tree with a number of links $> |V| - 1$.

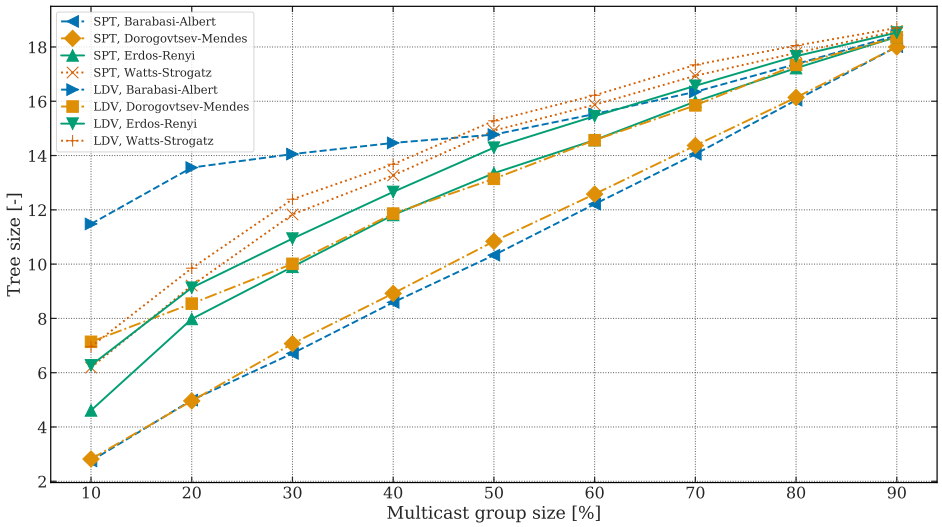


Figure 4.6: Effect of multicast group size on mean tree size at graph size $n = 20$.

Whereas the impact of multicast group size on the tree size is very easily identifiable, the investigation into the effect of network size was limited only to the window of three sizes: 10, 15 and 20 nodes. Although the range is not excessive the chart in Figure 4.7 indicates that the conclusions from previous perspectives were correct. The LDV formulation on the Barabási-Albert and Dorogovtsev-Mendes models tends to construct longer paths as the growing tree size suggests. The size of the LDV tree is almost two times larger than the SPT of those models. Interestingly, all curves seem to be linear in this detail.

Evaluation summary

The analysis of random graph instances using the proposed LDV model minimizing delay variation shows improvement compared to currently used approaches based on SPT. Interestingly, the results indicate that scale-free topologies with a higher number of links lead to lower delay variations. We assume that in closed network environments with specific traffic patterns, such as SA networks, it is possible to avoid high jitter values since

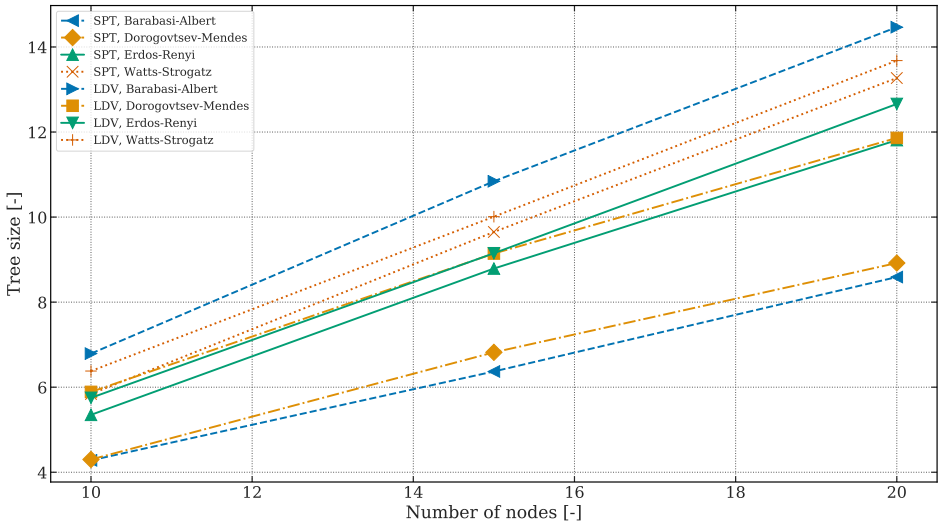


Figure 4.7: Effect of network size on mean tree size at the multicast group size of 20 %.

the local QoS mechanisms can be tuned precisely to fulfill the LDV goal by SDN or by traditional management tools.

It's difficult to be rigorous about whether a machine really 'knows', 'thinks', etc., because we're hard put to define these things. We understand human mental processes only slightly better than a fish understands swimming.

John McCarthy

5

Genetic Algorithm for an LDV Multicast Problem

As an exact optimization method based on the ILP does not scale well, the next step was to try an alternative approach. From the very nature of the $\mathcal{NP}$-complete problem, the goal was to adopt a method that can intelligently explore only promising solutions without searching throughout the whole solution space. It turned out that the most promising way was to employ the well-established GA emerging from the area of evolutionary computing.

Evolutionary algorithms can be seen as a subclass of a more general metaheuristic class. The formal definition of the term metaheuristic is shifting over time, as researchers have gone from a pre-theoretical period to a framework-centric [145] where the metaheuristics are commonly implemented or provided as a service. However, we can approach this from the definition proposed by Glover and Sörensen whereby "a metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop[1] optimization algorithms" [146]. As the same authors add in [110], metaheuristics are developed specifically to find a solution that is "good enough" in a computing time that is "small enough". As a result, they are not subject to a combinatorial explosion – the phenomenon where the computing time required to find the optimal solution of $\mathcal{NP}$-hard problems increases as an exponential function of the problem size.

[1] *heuriskein* (ancient Greek) and heurisricus (Latin): "to find out, discover."

The complexity, number of intricate operators and sometimes lack of rigorous testing of different implementations encouraged many critics recently. Moreover, the criticism was even more strengthened by the massive amount of metaphor algorithms inspired by nature, e.g., ant colonies, cuckoos, bees, bats, cats, wolves, and galaxy formation [147]. These reproofs for an insufficient validation and non-unified design even led to a call for standardization in [148]. Nevertheless, results achieved by metaheuristics on real-life problems proved the ability to succeed where other more rigid methods failed.

Generally speaking, metaheuristic algorithms seek the best solution from the solution space of the optimization problem. Potential solutions are evaluated, and various operators are employed to rapidly converge to the best solution. On the other hand, performed operations adjust possible solutions to avoid the local minimum and move the heuristic to the global one.

From the algorithmic perspective, the notion heuristic has been, for decades, a bit unclear and authors Romacya and Pelletier set out to define this heuristic against a historical backdrop in [149]. They conclude that "a heuristic is any device, be it a program, rule, piece of knowledge, etc., which one is not entirely confident will be useful in providing a practical solution, but which one has reason to believe will be useful, and which is added to a problem-solving system in expectation that on average the will improve". One of the most successful heuristic engines of all time is undoubtedly the human brain, a piece of great evidence indicating that the heuristic attempt to most problems in real-life is on average successful. As archaeological finds suggest, we can be grateful to evolution for having such powerful brains.

Evolutionary algorithms have been studied independently since the 1950s by several computer scientists. The goal was to use techniques from evolutionary systems and adopt them as an optimization tool. The backbone of evolutionary computation are evolutionary strategies, evolutionary programming, and genetic algorithms. The idea in all these systems was to evolve a population of candidate solutions for a given problem, using operators inspired by natural genetic variation and natural selection. Although researchers came with many evolution-inspired algorithms, John Holland was the first one who introduced a population-based metaheuristic algorithm with crossover, inversion, and mutation [150]. Holland published firm theoretical foundations of GAs in a ground-breaking book with the original goal to formally study the phenomenon of adaptation as it occurs in nature and how it might be exploited in computer systems. He proposed a theoretical framework which serves as the basis for most subsequent GA implementations.

5.1 Principles of Genetic Algorithms

The biological inspiration implicates the need to clarify some terms in the context of genetic algorithms. A very rough explanation is the following. Each cell of a living organism contains one or more chromosomes, strings of DNA, defining the organism. Each *chromosome* can be divided into *genes*, where each instance of a *gene* can have a different value or setting called *alleles*. The gene's position in the *chromosome* is the *locus*. All *chromosomes* together form a collection of the organism's genetic material called a *genome*. The particular set of genes contained in a *genome* is termed as the *genotype*. The physical interpretation of the organism's *genotype* is a *phenotype*. During reproduction, recombination, or crossover, occurs and genes are exchanged between each pair of *chromosomes* (*haploids*) to form a *gamete*. A mutation occurs when single *alleles* are changed from parent to offspring, typically as copy errors [150].

As Mitchel states in [150], Holland's orginal framework is an abstraction of biological evolution. The proposed GA is a set of methods to move from an initial population of *chromosomes*, strings of bits, to a newly created population using a sort of natural selection, and genetics-inspired operators of crossover, mutation, and inversion. In each generation, the selection operator chooses *chromosomes* to reproduce, where the fitter *chromosomes* produce more offspring. The crossover operator exchanges subparts of two *chromosomes* and the mutation operator randomly changes the *allele* values of some location in the *chromosome*.

5.1.1 General structure

Generally, a GA consists of five components as described in [151].

1. An encoding method that is a genetic representation (*genotype*) of solutions to the program.
2. A way to create an initial population of individuals (*chromosomes*).
3. An evaluation function, rating solutions in terms of their fitness, and a selection mechanism.
4. Genetic operators (crossover and mutation) that alter the genetic composition of offspring during reproduction.
5. Parameters of the genetic algorithm.

The general structure of the GA shown in Algorithm 2 comes from Holland's original proposal. The framework is reusable, but the implementation of genetic operators and parameter tuning is often unique for each problem. Although a GA can deliver very quickly feasible solutions, almost every component, or algorithm step, contains critical points that can negatively influece the solution quality or the convergence speed if set up

improperly.

Algorithm 2 General structure of GA

$t \leftarrow 0$
initialize population $P(t)$ for generation t
evaluate population $P(t)$
while termination condition $= false$ **do**
    yield offspring $C(t) \leftarrow$ recombine $P(t)$ from randomly selected individuals
    evaluate $C(t)$
    yield next generation $P(t+1) \leftarrow$ apply genetic operators on $P(t) \cup C(t)$
    $t \leftarrow t+1$

Since the GA is a known metaheuristic framework, the following text will only briefly describe the purpose of each GA's component mainly focusing on implementation details used in this thesis. More details about GAs can be found for example in [150].

5.1.2 Components

The following sections describe the function and features of components used in a GA, their context in the evolution process and specific implementation variants chosen for the LDV problem.

Encoding

Firstly, one of the most important decisions that can considerably affect performance and time complexity of the GA is the choice of an appropriate encoding/decoding method of an internal problem representation, i.e., the transformation from the problem instance to its *genotype*. The encoding method should meet the following properties as summarized in [151].

1. Non-redundancy - one-to-one mapping between an individual and solution.
2. Legality - any existing permutation of an encoding is a valid solution.
3. Completeness - any potential solution in a solution space has a corresponding encoding.
4. Causality - in terms of a solution locality a small change to the genotype implies a small change to the phenotype.
5. Ease - going from genotype to phenotype and vice versa should be easy enough.

6. Short and low order schemata[2] - forces the GA to the best chromosomes.
7. Lamarckian property - the meaning of alleles for a gene is context-independent.
8. Unbiased representation of trees - the number of individuals representing a solution is identical for all solutions.

Even though many schemes related to the Steiner tree problem (Steiner node vector, Predecessors, Connectivity matrix of edges, etc.) were published, the proposed solution adheres to a traditional binary encoding which is the simplest. The binary encoding is created by a string of bits and is called a characteristic vector. The characteristic vector $[x(1), x(2), ..., x(|L|)]$ is fixed-length and features fixed-order mapping where each vector index is associated with exactly one graph link $\ell \in L$. The value of $x_\ell \in \{0, 1\}$ is one if the given link is part of the solution or zero otherwise.

Binary encoding does not pose a good option when attempting to solution search using a random string generation. Especially in large instances, the probability that a tree, moreover a feasible tree, will be generated is tiny. One needs to employ a different approach, where for example binary encoding is a simple method of transporting graph context between generations, but other graph related operations are performed holistically on a graph-aware data structure. In the case of our implementation, we built on top of an adjacency list representation implemented by a NetworkX library. It is evident that the transformation of data structures can limit the overall performance. However, the implementation simplicity and fulfillment of the required properties (except legality) makes the characteristic vector encoding of choice.

Population

In the context of a GA, the population is a set of individuals containing a specific configuration of alleles in genes. The individual's configuration is an input for a fitness function expressing the value of the individual in the given generation. Depending on the GA framework the population can be of a fixed size or variable size with or without boundaries. Fix-sized populations are more predictable in terms of the consumption of computation resources, as the evolution of the population throughout the generations remains constant.

Solution convergence from the initial to the final population needs to be reasonably

[2]The notion *schemata* (or *schemas*) was initially introduced by Holland to formalize the concept of *building blocks*, explaining the idea that good solutions, reached by the evolution process in GA, are built from good *building blocks*. A schema H is a set of bit strings composed of ones, zeros, and asterisks representing wildcards and it is defined by a number of defined bits, i.e., order $o(H)$, and defining distance $\delta(H)$ between the outermost defined bits. Using these features and some other features of *schemata*, Holland proposed an inequality describing evolutionary dynamics known as the Schema Theorem [152]. As Mitchel comments in [150] "the Theorem is often interpreted as implying that short, low-order schemata whose average fitness remains above the mean will receive exponentially increasing number of samples over time."

progressive. A very fast, or premature, convergence, may lead to solutions stuck in a local minimum whereas a slow convergence may generate even an unfeasible solution in the given evaluation period. To determine the level of convergence is not a clear task, but a common practice is to consider the share of converged alleles in the population. The allele is converged when more than 95 % of the population share the same value for a specific *gene* [153], or the allele is considered lost if the value is identical at all individuals. The converged alleles effectively reduce the solution space available to the crossover operator. Premature convergence is related to population diversity[3] and, consequently, to the variance of fitnesses in the population [150]. Methods designated to preserve population diversity and their input parameter tuning are described further in the chapter.

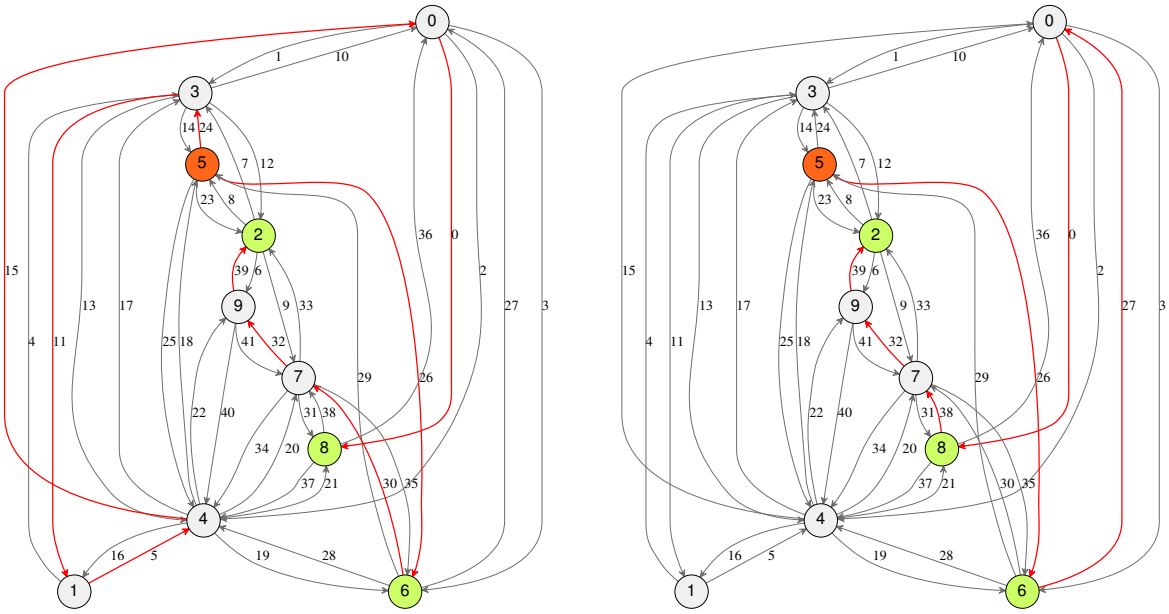
Besides the optimization of the GA's input parameters, an initial population plays a key role. An adequately generated initial solution has the potential to improve convergence speed while targeting a high-quality solution significantly. Simply said, the purpose of a GA is not to reinvent the wheel but to tune it to its best performance. On the other hand, the initial population has to provide an adequate level of entropy. A population containing too similar individuals slowes convergence speed as randomness is missing to escape a local minimum.

A generally accepted hypothesis is that individuals in the initial population should be distributed as uniform as possibly in the search space. Although uniform distribution is easily achievable in case of binary encoding used at the characteristic vector, it is not so straightforward when we want to obtain an individual representing a tree. Seeking a valid solution even in a solution space split into subgroups would be extremely exhausting when we consider the number of combinations equal to $2^{|L|}$. Accepting these arguments, one can expect that a problem-aware approach to the generation of an initial individual should deliver a higher-quality population.

Two different strategies demonstrated in Figure 5.1 were implemented for a construction of the initial population. The first one follows an approach of a randomized depth-first search algorithm published originally in [155] and it is used for a random multicast tree construction in many publications, e.g., a modified version in [151]. Our implementation follows the common randomized Depth-First Search (DFS) algorithm. The algorithm constructs a multicast tree beginning from the publisher node v_p towards all terminals by progressively visiting undiscovered nodes. At each node, the algorithm randomly chooses from a set of output links and the rest of links is saved to a backtrack log to avoid loops and to be able to reconstruct the tree.

The second algorithm, let us term it a snake algorithm, had as its goal to design a

[3]The term diversity indicates dissimilarities of individuals in the population. Without a diversity-preserving mechanism, there is a risk of the best individual taking over the population before the fitness landscape is appropriately explored [154].



(a) Initial tree created by randomized DFS algorithm. The algorithm forms trees with more branches.

(b) Initial tree created by the snake algorithm. The algorithm tends to generate a multicast tree in the form of a path.

Figure 5.1: Differences between initial trees found by randomized DFS 5.1a and the snake algorithm 5.1b on a small instance of the Barabási-Albert model.

different approach to generate a tree to validate the previously described DFS algorithm. At first, a set of Steiner nodes $M \subset V$ is initiated. To fill the set, the algorithm selects multicast terminals to be a Steiner node[4] with a probability $\beta = 0.6$, and for each nonselected multicast terminal, the algorithm tries to choose, with a probability $1 - \beta = 0.4$, a random node from the rest of graph nodes $V \backslash (S \cup \{v_p\})$. In the second stage, the snake algorithm split set S into equal-sized disjunctive subsets where the number of subsets is equal to the number of left multicast terminals. Then, the algorithm iterates the left terminals and each time it constructs a path connecting all Steiner nodes in the given subset and the terminal at the end of the path using the shortest path algorithm. At each iteration, the previous terminal is used as a starting node, a Steiner node, and the construction follows the same schema of building another path to the next terminal. The tree is always built from the multicast group publisher v_p. The snake algorithm tends to produce trees in the form of a long path with a multicast terminal at the end. Although the algorithm does not always deliver a valid multicast tree, as during the creation a loop may appear, it leads to unusual tree compositions feasible for evaluation purposes.

As a preliminary evaluation indicated, any of the proposed strategies is superior, and so, both procedures were employed in the final implementation of the proposed GA. The goal of this hybrid strategy is to widen the potential spectrum of unique individuals in the

[4]Since each node represents an Ethernet switch it can forward multicast messages not only to the directly attached IED but it can pass the message to another connected switch as well.

population. The former or latter procedure generates each individual in the population with equal probability.

Fitness function

A fitness function is the Alpha and Omega of every GAs. The evolution process is almost exclusively driven by the fitness function that is used to express the quality of each individual, and so, its value for the next generations. Using a biological analogy, individuals adapt to the environment set up by the fitness function. As Kinnear notes in [156] "the fitness function is the only chance that you have to communicate your intentions" and "make sure that it communicates precisely what you desire." The fitness function has a major impact on how the population will evolve, thus, how the individuals will change over time. The adequately designed fitness function is a critical part of a GA as it influences the selection strategy and so the selection pressure as well.

The function should be designed with the concept of partial credit in mind. In case the fitness function is too narrow, i.e., the area where an individual is highly valuated is only in a short range of possible solutions, the algorithm may not converge at all. The concept of partial credit allows individuals to be compared and distinguish more or less successful solutions. For proper convergence of the algorithm, the fitness function needs to provide such a score suggesting a promising direction in the evolution. This approach is in contradiction to decision problems in computer science focusing only on the simple task of whether a solution was found or not [156].

The evolution process needs to be informed with a proper granularity whether the population is approaching an optimum or not. Higher granularity usually means the function is not steep and the evolution process can converge to the optimal solution more steadily. It also keeps a better diversity of individuals in the population. If a few members of the population, the elite, have very high fitness in comparison to others, more fit individuals would quickly dominate and result in premature convergence [157].

To formalize an intuitive way the fitness function is evaluated, it is convenient to mention that a GA on binary encoding can handle only pseudo-Boolean optimization problems directly $f_\Phi : \mathbb{B}^n \to \mathbb{R}$, i.e., fitness function $\Phi = f_\Phi$. An example of such a problem is the OneMax[5] problem. Other problems of the general form $f_\Phi : S \to \mathbb{R}$ require a decoding function for a transformation of an individual to the problem space S. For this reason, it is useful to distinguish between the decoding function $\Psi : \mathbb{B}^n \to S$ and the objective function $f_\Phi : S \to \mathbb{R}$, so that $\Phi = f_\Phi \cdot \Psi$ [158]. As the LDV problem

[5]The OneMax problem is a well-studied problem with unimodal fitness landscape. For a binary string x of length l, the objective is to maximize $\sum x_i, x_i \in \mathbb{B}$. The OneMax problem is often used for the benchmarking of GAs.

requires multiple mathematical operations to be done on the candidate tree[6], i.e., decoded individual, decoding has to be employed in this case. The notion fitness function is tallied with the objective function in the following paragraphs.

Unconstrained optimization problems come from the objective function when formulating the fitness function. In case of constrained optimization problems, an exterior penalty can be applied to convert the constrained problem to the unconstrained one virtually. The exterior penalty function keeps the algorithm in the area of feasible solutions. Since the exterior penalty is effective only outside of the feasible region, it finds only extremes occurring behind the region boundaries. On the other side can be employed an interior penalty preventing the individual from reaching the boundary of the feasible region.

The fitness function (5.1) designed for our delay constrained LDV problem is composed of four parts. It is inspired by the fitness function published in [151], but it is simplified for the bandwidth and cost part. On the other hand, as the proposed solution does not employ a tree recovery technique under all conditions, the fitness function reflects the infeasible and invalid solution by additional penalties. The combination of exterior penalties effectively creates gaps where the individual hops between function levels depending on input coefficients. Two coefficients α and γ are employed by the fitness function. While α is used to scale the score properly, coefficient γ is a degree of penalty. In both cases, coefficients remain constant during the algorithm run. The fitness function was numerically evaluated, and we decided to put $\alpha = 10$ and $\gamma = 0.5$.

A core part (5.2) of the fitness function gives an actual quality of the individual depending on the reached delay variation. The penalty expressed in (5.3) reflects multicast tree completeness, i.e., the number of terminals reachable from the root. The penalty (5.4) restricts the creation of loops and prefers valid trees. The last penalty described in (5.5) favors solutions not breaking the delay constraint.

[6]For example to compute end-to-end delays between the publisher $v_p \in V$ and all subscribers $S \subseteq V \backslash \{v_p\}$ using node and link delays, finding maximum and minimum values and compute δ_T

$$f_\Phi(T(v_p, S))) = (f_b + p_c) \cdot p_t \cdot p_l \tag{5.1}$$

$$f_b = \alpha \cdot \left(\frac{1}{\delta_T + \alpha} + 1 \right) \tag{5.2}$$

$$p_c = \frac{|R_T(v_p, S)| - |S|}{|S|} \cdot \alpha \tag{5.3}$$

$$p_t = \begin{cases} 1 & \text{if } T(v_p, S) \text{ is branching} \\ \gamma & \text{otherwise} \end{cases} \tag{5.4}$$

$$p_l = \begin{cases} 1 & \text{if } \delta_{T_{max}} \leq \Delta \\ \gamma & \text{otherwise} \end{cases} \tag{5.5}$$

$$R_T(v_p, S) = |\{|P_T(v_p, S)| > 0 | \forall s \in S\}| \tag{5.6}$$

$$\tag{5.7}$$

To make clear the fitness function defined above, we remind and detail the meaning of all functions.

- $T(v_p, S)$ is a sub-graph of G compounded of a multicast source node (publisher) $v_p \in V$, and multicast destination nodes (subscribers) $S \subseteq V \backslash \{v_p\}$ where the set $S \cup \{v_p\}$ is called the multicast group
- $P_T(v_p, s), s \in S$ is a set of links $\ell \in L$ on a path from node v_p to node s in the tree $T(v_p, S)$
- $R_T(v_p, S)$ defined in (5.6) is the number of reachable subscribers in S from node v_p in the tree $T(v_p, S)$

Although the LDV problem has a unimodal fitness landscape as the OneMax problem mentioned above, the nature of both problems predefines the smallest acceptable change on the chromosome affecting a step size in the fitness landscape. While the OneMax problem can be modified with same weight on a single bit in the string, i.e., switching one bit always changes the objective value by $\pm$ 1, in case of the LDV problem it is possible to traverse from one valid solution to another by changing at least two bits (two links), but possibly more. The nature of the LDV problem with unequal weights and how δ_T is computed may lead to big jumps in the fitness landscape. Moreover, two reasonable solutions do not need to contain *building blocks* of the optimal solution, as in the case of OneMax problem. Even though these features are not encouraging and

place high demands on the preservation of the population diversity, we show that from the perspective of multimodal optimization[7] GA is an acceptable way.

Selection

Having the initial population generated and the fitness function defined, the next step is to choose an appropriate selection method. Considering the evolution process, the GA can be split into two behavioral domains: the exploitation and exploration of the search space. In the exploitation phase, the selection is performed on a given population with an objective to choose a sample of individuals predetermined to create an offspring forming the next generation of the population. Generally, selection changes the population variance, and in order to avoid premature convergence, the variation operators have to counteract that effect in the exploration phase [159].

The selection method has to emphasize the fitness of individuals and properly prioritize the fitter ones. In this context, the term properly is strongly related to the role of elitism in the selection pressure described above. The method needs to be well-balanced concerning the diversity of an impending generation. An excessively intensive selection sidelines individuals with lower fitness in favor of the elite, which takes control of the population rapidly and does not allow the evolution process to discover possibly better solutions. On the other hand, an overly careful selection slows the evolution and may not reach even a suboptimal solution in a given number of generations.

Although a vast number of advanced selection methods have been published, the method of choice is a simplified tournament selection. This method is easy to implement, and it scales well as it can be parallelized. While the original tournament method incorporates a random factor used to decide whether to choose a fitter individual from two individuals randomly chosen from the population, the simplified tournament method randomly chooses a group of individuals of a given size and selects the fittest one. In our case, this is done population-size times keeping the fixed-size of the population. Both methods allow an individual to be chosen as a parent repeatedly. The original method is influenced by the random selection of two individuals and a given parameter, and the simplified method is controlled strictly by the tournament group size: the bigger the group the higher the selection pressure. The size of the group chosen for our delay constrained LDV problem is described further.

[7] In the last chapter, a selection of good valid solutions is eventually expected to be used for the optimization of the multi-tree BDLDV problem.

Elitism

As the selection process is random-based, it may omit even the best individual causing the loss of a schema with high potential. Moreover, these highly valuated individuals may be destroyed by crossover or mutation. To overcome this situation, De Jong introduced *Elitism* in [153] as an extension to selection methods forcing the GA to keep a defined number of the best individuals at each generation. Elitism can significantly improve the GA's performance [150]. In our implementation, elitism keeps the single best individual at the generation, and this feature is a part of the input parameters.

Crossover

The crossover operator is considered to be a significant instrument of variation in a GA [150]. The basic idea behind the recombination is to add innovation to the evolution process. To achieve this, the crossover operator should be able to make significant jumps in the search space, while preserving the legality of the solution as much as possible. Mating the parents to obtain an offspring is the distinguishing feature of a GA that the realiziation of such a coarse operation. The crossover operator should lead to as much diversity in the offspring population as possible [159].

Unfortunately, there is no definitive guidance on what type of crossover operator to use and how to tune it. Traditionally, two simple methods, while often modified, are used to produce an offspring: single-point and two-point crossover. The principle of both relies on a random selection of mating points, where parents' chromosomes are cut to segments, and these are exchanged with each other. At the single-point crossover, schemes with long lengths are likely to be destroyed, and it always preserves endpoints when exchanging the segments, i.e., preferring some *loci* during the evolution [150]. The two-point crossover operator partly solves the reprehension.

The methods described above are applicable on bit-string encoding schema meaning they are extremely fast. However, there is no guarantee that the product of the crossover operation will be a feasible solution, preferably a multicast tree. Although the first idea could be that the selection operator takes care of illegal offspring in the next generation, results show that passing valid individuals to the next generation is beneficial. As the authors of [151] show on simulation results, employing a one-point crossover operator in conjunction with a fast check and tree recovery algorithm leads to faster convergence then crossover operators specially designed for multicast trees. A key concept is the recovery algorithm which tries to reconstruct damaged individuals. Potentially valuable *building blocks* remain preserved between generations, and the evolution process does not suffer from a slump in the population quality.

In the proposed GA implementation, two entirely different mating strategies were selected. The first is the already mentioned one-point crossover inspired in the *Crossover II* scheme proposed in [151]. The one-point operator is directly applicable to our encoding scheme, and it could be simply imported from the DEAP framework [160] we used to build the presented GA. In the case of the two-point operator, a preliminary evaluation indicated that this mating strategy caused too much disruption of the offspring resulting in profoundly broken trees, so the operator was rejected.

The second crossover operator outlined in Algorithm 3 was designed with the objective to minimize δ_T in mind. The operator is noted besttree, and it is an opposition to the one-point crossover in terms of complexity. To be able to reflect the objective, the operator is relatively complex as the decoding Ψ from *genotype* to *phenotype* has to be performed. This operator is constructed to be objective-aware with the goal to construct the two best while dissimilar multicast trees act to preserve population diversity. From the local perspective at the time of the offspring construction, only the most promising *building blocks* are utilized. As a variation operator performed in the exploration phase, it is in contradiction with the requirement to not bias the population stated in [159], meaning that such an operation should not change the mean fitness of the population. This approach may be a potential disadvantage in the evolution process as there is an intersection with a function of the selection operator, and so population diversity may be endangered.

The algorithm consists of several loops and a modified Breadth-First Search (BFS) to find all simple paths where a single path can be found in $\mathcal{O}(|V| + |L|)$. The number of simple paths may be huge in a graph with a high number of links though. In practice, the number of paths is limited as the evaluated graph is combined from the parents that are most likely trees as well. Similarly, the while loop is not, on average case, iterated till the δ_T array is empty as two offspring are found earlier. Although matrix computations are reasonably fast, the potentially limiting part of the algorithm is memory consumption as the number of subscriber pairs, given by combinations without repetition, is multiplied by the number of simple paths to each subscriber. However, a practical evaluation on dense instances with $|V| > 25$ and $|L| > 100$, i.e., instances larger than in 4.2.2, did not encounter any scalability issues.

No-Genotype Duplicates

As the preservation of population diversity is an essential key to exploring the search space effectively, we implemented a simple diversity-preserving mechanism. The very intuitive way to enforce diversity is not to allow *genotype* duplicates. This simple algorithm was published in [161], and the idea is to prevent identical individuals from remaining in the population as a natural way of ensuring diversity. Although the mechanism is straight-

Algorithm 3 Besttree crossover strategy

```
t ← Ψ (parent1 bitwise or parent2)
ap ← find all simple paths by BFS on t from the publisher to all reachable subscribers
for p ← ap do
    e2ed[p] ← compute end-to-end delay for p
for sp ← subscriber pairs do
    δ_T[sp] ← compute matrix of δ_T for all sp simple paths
sort δ_T[] in ascending order
while size of offspring < 2 and size of δ_T[] > 0 do
    cd ← pop δ_T[]
    cdsp ← subscriber pair for cd
    mine2ed ← lower e2ed for cd
    maxe2ed ← upper e2ed for cd
    for rs ← reachable subscriber not in cdsp do
        if maxe2ed ≥ e2ed[rs] ≤ mine2ed then
            offspring[i] ← randomly select valid path for rs
        else
            break
    delete δ_T[cd]
if size of offspring < 2 then
    offspring ← back off to one-point for missing offspring
return Ψ(offspring)^-1
```

forward to implement, and it is efficient due to the bit-string encoding, it may not be powerful enough as shown in [154] on the TwoMax[8] problem. On the other hand, advanced mechanisms would require decoding to the *phenotype* as in the case of the besttree crossover operator.

Mutation

From the beginning of the GA era, the major variation operator was considered to be the crossover operator delivering a sufficient level of innovation among individuals in the population. The mutation operator played only a minor role ensuring that the population avoids a permanent fixation at any particular *locus*. The perspective was changed by Spears in [162]. The author provides a justification for Holland's construction theory where the role of crossover is to build high-order *building blocks* from low-order ones. The mutation, on the other hand, can provide a higher level of disruption. Nonetheless, the author, concludes, based on experimental results, that the distinction between crossover and mutation may not be necessary. The debate on this topic continues unabated even after a decade as shown in [163], where the author of the survey concludes that both

[8]The function TwoMax is essentially the maximum of OneMax and ZeroMax. Local optima are solutions composed of 0n and 1n where the number of zeros or the number of ones, respectively, is maximized [154].

crossover and mutation are necessary.

Since the random bit flip in the *chromosome* is ineffective, as in the case of the population construction, a new mutation method was designed. The idea behind the algorithm is straightforward. At first, a multicast subscriber is randomly chosen from a set of subscribers. Then, all input links and links on the branch to the subscribing node are removed, a new input link is randomly selected and, using DFS for backtracking the tree, a mutated branch is constructed. As mutating all subscriber branches may be overly intensive from an evolution perspective, a dedicated terminal selection probability is employed beyond the standard mutation probability. The second probability parameter influences the total amount of subscribers selected for the mutation. The algorithm's worst-case performance complexity is $\mathcal{O}(|V| + |L|)$.

An extended version of the mutation algorithm incorporates an objective-aware selection of branches suitable for reconstruction. In this case, a similar mechanism to the besttree crossover operator is implemented, but oppositely. A higher mutation priority is assigned to tree branches leading to subscribers with the largest δ_T. Similarly, the requirement to not bias the population is not fulfilled though.

During the evaluation of the proposed mutation method, it turned out that in the composition of presented components the mutation operation is indeed a significant player when considering population diversity. From that point, the goal was not to design any new mutation algorithm but to focus on the control of the mutation intensity during the evolution process.

Apparently, the most common approach is a fixed-level probability where the level of the probability does not change during the evolution process. Although this approach is simple and quite predictable, it may oversee some population related states. For example, the population is, at the beginning of the evolution, dissimilar enough and needs time to evolve robustly. In this phase, a too strong mutation could disrupt individuals with a fruitful *genome* that was not yet able to spread sufficiently in the population. The population needs to stabilize at first. On the other hand, the population may be stagnating in local minima in later stages of the evolution, and a weak mutation would not deliver enough innovation. To address these and other tasks we propose methods on how to control the mutation intensity:

Constant mutation fixed mutation probability during the complete evaluation process. As Spears notes in [162] the highest construction potential is at p_m=0.5.

Intensified mutation the mutation probability is attenuated during the evaluation process with a geometric progression. The goal is to gradually achieve more significant mutation probabilities after 70 % of the generations were evaluated. The idea comes

from the conclusion presented as well by Spears in [162] that the crossover has higher constructive levels until the population is 70 % converged.

Adaptive mutation with feedback a self-adaptation method reflecting a population's diversity to the mutation probability. Self-adaptation mechanisms are studied from the very beginning of the evolution algorithms as reviewed in [159]. The well-known representative is Rechenberg's 1/5th rule[9], however, our goal was to design a method considering population diversity. While the previous two methods have predefined behavior during the evolution, the proposed self-adaptive method takes into account mean Hamming distance of individuals in the current population. It tries to keep the mean Hamming distance at a desired level by modifying the mutation probability for the next generation. The mutation change indicator σ_m shown in equation (5.8) reflects a ratio of the population's mean Hamming distance Δ_{ewma}, smoothed using the exponentially weighted moving average with a window of 5 generations, to a given distance limit Δ_{limit}. Assuming some probability p_m^g is given at the start of the evolution, a probability for the next generation p_m^{g+1} is computed using the equation expressed in (5.9).

$$\sigma_m = 1 - \frac{\Delta_{ewma}}{\Delta_{limit}} \tag{5.8}$$

$$p_m^{g+1} = \begin{cases} \sigma_m(p_m^g + 1) & \text{if } \sigma_m < 0 \\ p_m^g + \sigma_m(1 - p_m^g) & \text{otherwise} \end{cases} \tag{5.9}$$

Tree recovery

It turned out that the tree recovery part of the evolution process outlined in Algorithm 4 is vital. The recovery helps to turn illegal solutions into a valid one, and so the selection and crossover are more successful in the next generation, as a valid solution naturally tends to produce another valid solution. It is obvious that the recovery operation requires decoding the *phenotype*. Nonetheless, decoding has to be performed on at invalidated individuals to compute new fitness and check legality as well. The recovery algorithm is performed after all other variation operators are applied. The algorithm's worst-case performance complexity is $\mathcal{O}(|V||L|)$.

As all instances were generated with bidirectional topologies, the recovery algorithm has to handle situations like short loops when the offspring contains loops between neigh-

[9] As explained in [159], Rechenberg's proposed in [164] a 1/5th rule. It relies on counting the successful and unsuccessful mutations for a certain number of generations. If more than 1/5th of mutations lead to an improvement the mutation strength is increased and decreased otherwise. The aim was to stay in the so-called evolution window guaranteeing nearly optimal progress.

boring nodes. The recovery algorithm is intended to produce a nearly legal solution meaning that for the sake of performance some recovery operations may produce both illegal and infeasible individuals. Such faulty individuals are the subject of the following mutation or improvement steps, and potential illegality is addressed in the fitness function. The evolution process an suppresses invalid solution in the selection step. The algorithm pre-evaluation indicated that a successful recovery rate is highly above 90 %.

Algorithm 4 Tree recovery

```
o ← Ψ(individual to recover)
remove ingress links to the source node
for n ← node in o do
    remove short loops on bidirectional links
    remove more inputs to n
    trace long loops and remove the last link preceding an already visited node
t ← orphan terminals not part of o
for n ← node in t do
    apply DFS from n to find the first node being a member of o tree
l ← terminal leaves of o not being a multicast terminal
for n ← node in l do
    clear the path from n to the first branching node
```

Local improvement

Although the great power of the purest GA is both blind and effective in looking for a global optimum in the search space, it may be at a competitive disadvantage to problem-aware algorithms. Local search algorithms miss the generality of GA, but often exploit the context and provide an acceptable local optimum promptly. Combining problem-specific information with GAs gave rise to hybrid schemes [165]. A hybrid technique builds on the local optimization of the objective function, and thus, it incorporates a local improvement operator to the evolution process. The new operator tries to increase the fitness of the individual using problem-specific information. However, it should preserve the population's diversity.

From the biological perspective, the hybrid technique is a kind of interaction between the evolution and learning processes. The role of learning, more generally *phenotypic* plasticity, in the evolution theory was the subject of intensive research over the last two centuries. It began with a Lamarckian hypothesis stating that traits acquired during the lifetime of an organism can be transmitted genetically to the organism's offspring. This concept was rejected as the evidence shows there is no direct way how to reverse transcript the trait to the individual's *genome* [150]. However, the effect of learning on evolution can

be significant but more indirect, as suggests the Baldwin effect[10] or the more plausible mechanism called genetic assimilation[11] [150].

The local improvement algorithm in our proposed implementation is straightforward. The basic idea behind the algorithm is to identify branches of the multicast tree that introduce the highest delay variation, i.e., branches with maximum and minimum end-to-end delay, and adjust them to lower the variation. The selected branches are invalidated, and a modified DFS algorithm is applied to construct a new path in the graph from the node where the tree is branching. Although this approach does not guarantee any improvement, it can reasonably quickly shift the individual to the local optimum. In case the individual was improved, the improvement is directly encoded in its genome. This approach complies with the Lamarckian hypothesis and it is in contradiction to real genetics as presented above.

5.2 Evaluation of a GA on the LDV multicast problem

At this point, when the background of the proposed GA framework is known, the next step is to evaluate it. As is evident from the previous sections, the GA can swell into an extremely complex structure, and so, the evaluation had to be split into several levels. Even though the most important goal of the evaluation is to verify how well the algorithm can deliver an optimal solution, another essential part of the evaluation is to identify optimal input parameters. As shown in the following sections the combination of input parameters is crucial and dramatically affects the quality of results.

As the solver for the LP approach is parallelized, we attempted to parallel the GA implementation as well. The initial idea was to run the algorithm simultaneously on a set of instances, each computed in one thread, i.e., instance-level parallelization. Such an approach seemed to be acceptable for both the algorithm evaluation and generation of initial configurations employed in the next chapter. Nevertheless, the asymmetrical complexity of instances led to blocking states when smaller instances had to wait for larger instances to be finished. As the utilization of available computing resources was, in some cases, exceedingly ineffective, a second version was implemented using individual-level parallelization. Even though individuals in the population are asymmetric in complexity as well, the micro-blocking states were insignificant, and so, the resource utilization became more effective. The main challenges with implementation lay in the combination of

[10] As summarized in [150], the capacity to acquire a specific trait allows the learning organism to survive preferentially, thus giving genetic variation the possibility of independently discovering the desired trait. Without learning, the likelihood of survival decreases.

[11] In short, if organisms are subject to environmental changes and genes for particular traits are in the population, they can be expressed relatively quickly, especially if the acquired phenotypic adaptations have kept the species from dying off [150].

directed graphs and the level of randomness in the GA since every genetic operator needs to take care of loops in the graph.

5.2.1 Evaluation process

All genetic operator and routines described in Section 5.1.2 were naturally validated throughout the implementation of the GA to ensure each step in the evolution process worked smoothly. The continuous sub-evaluation led to an iterative development progress where a particular deficiency in the general approach brought a new method into being. Eventually, the scope of implemented GA went beyond the possibility of grid evaluation. The Cartesian product of all input parameters became enormous. Although the evaluation was initially done in a grid manner, the advanced versions made this way of evaluation nearly impossible in an acceptable time. All computations were done on the National Grid Infrastructure (NGI) operated by MetaCentrum. Even though the computation resources in NGI are enormous, we had to employ different and more goal-aware methods.

The first one is a way of pre-defined scenarios where we not only obtain a significant amount of data from the parameter search space, but we can compare implemented operators to each other under specific conditions, i.e., to observe the impact of a particular operator. Initially, a set of 30 scenarios was designed to cover the desired space. Another 5 scenarios were added afterward by the basic scenarios to detail particular areas. All evaluation scenarios are described in Appendix A.1 of this thesis.

When considering alternatives to the grid search, we inspired ourselves in methods used in machine learning and adopted a Bayesian algorithm for hyperparametric optimization. Since this method is not valuable for the isolated evaluation of input parameters, the optimization methods are further explained in Section 5.3.

A set of instances was created to be able to evaluate operators and input parameters consistently. This set is composed of random graphs generated using the Barabási-Albert model as described in Table 4.1 in the previous chapter. The purpose of the set is to capture a sufficient diversity of instances with such complexity that the LP method is still able to deliver an optimal solution in an acceptable amount of time. The set contains 6 instances composed of 15, 20 and 25 nodes with 30% and 60% multicast coverage. The number of links in the instances varies from 70 to 130 which is a reasonable amount in the context of this thesis.

Evaluation criterion

Without a doubt, the ultimate goal of the proposed algorithm is to deliver an optimal solution which means the primary criteria when evaluating the set of instances is a ratio of

optimal solutions *Opt* found for a given configuration, or scenario. To obtain a statistically significant sample, the set of instances was computed 10 times for each configuration. As noted above, the optimal solutions were obtained using the LP model described in Section 4.2.1. Having the optimal solutions, we can rely on exact numbers when comparing methods, though only for instances of limited size.

Since limiting the evaluation of an approximative algorithm only to the ratio of optimal solutions is short-sighted, other metrics were calculated. The first are relative quality and relative error expressed in (5.10) and (5.11). The best-achieved delta variation by GA is denoted as δ_T^{ga} and the exact solution as δ_T^{opt}. If $R = 1$, and thus $\epsilon = 0$, the solution is optimal, else there exists some level of discrepancy between the exact and approximation solution.

$$R = \frac{\delta_T^{ga}}{\delta_T^{opt}} \qquad R \in \langle 1, \infty) \tag{5.10}$$

$$\epsilon = 1 - \frac{1}{R} \qquad \epsilon \in \langle 0, 1\rangle \tag{5.11}$$

Although the previous metrics reflect the quality of solutions for a given configuration, there are other criteria to take into account when choosing the proper input parameters. We designed a special indicator named the performance index P_i to be able to compare the performance of different configuration combinations. Indicator P_i defined in (5.12) considers the number of multicast subscribers $|S|$ and links $|L|$ concerning relative quality R and total evaluation time t_e^{ga} for a given instance. This metric helps to understand how the particular GA configuration is performing while taking into account the instance size.

$$P_i = \frac{|S|\,|L|}{R^2 t_e^{ga}} \tag{5.12}$$

Since the original evaluation times were available for optimal solutions coming from the LP model, we can easily obtain latency speed up S_l defined in (5.13). The speed up simply express the ratio between execution times of the optimal LP model as t_e^{opt} and GA t_e^{ga} for a given instance. Both times are normalized per CPU. Even though the ratio is only a coarse indicator of performance, it can be a helpful metric under the condition that both computations are run on the same platform. In the case of $S_l \geq 1$, the GA performs better than LP.

$$S_l = \frac{t_e^{opt}}{t_e^{ga}} \tag{5.13}$$

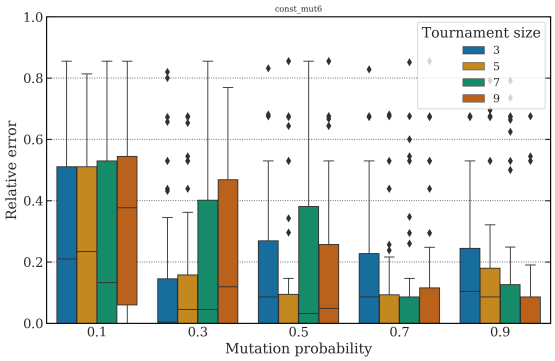
5.2.2 Primary operators

Initially, the evaluation section focuses on primary operators known from the traditional GAs, and after that, we continue with less common operators trying to enhance the rigid evolution schema.

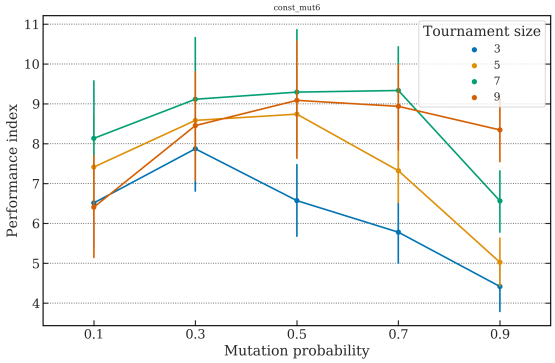
Selection

As the selection is the first part of the evolution process, we start the evaluation with it. In general, results obtained from evaluation scenarios show that the larger the tournament, the smaller the ϵ. This statement is valid for the ratio of optimal solutions achieved as well. From the perspective of other input parameters, tournament size shows sensitivity to the level of mutation probability as one can see in Figure 5.2a. The positive impact is most noticeable for sizes above 3 and higher mutation probabilities. Other input parameters do not show a significant dependence on tournament size in terms of solution quality.

In all scenarios, a bigger group of individuals randomly selected for the tournament leads to better performance as depicted in Figure 5.2b. The difference in P_i between the sizes, especially at high mutations rates, is evident. Taking into account lower relative errors and higher performance, the proposed GA should preferably use a tournament size close to 9.



(a) Relative error achieved for different tournament sizes and mutation probabilities. Larger tournaments give better results under higher mutation rates.



(b) The reported P_i for different tournament sizes and mutation probabilities. The smallest tournaments achieve the worst performance under high mutation rates.

Figure 5.2: Impact of tournament size on quality of results considering a changing mutation rate for the constant mutation (scenario const_mut6).

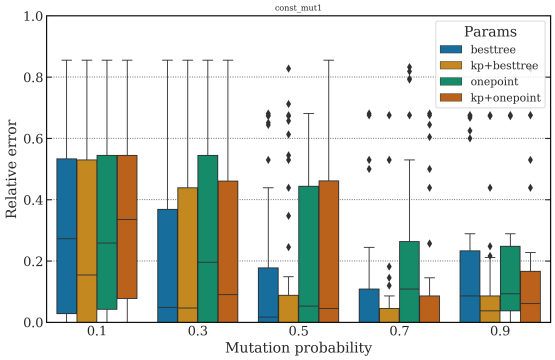
Crossover

Two crossover methods were employed in the comparison as described in 5.1.2, the standard one-point mate and the newly proposed besttree method. Scenarios with the constant and adaptive mutation were chosen to compare these methods. As one can see in

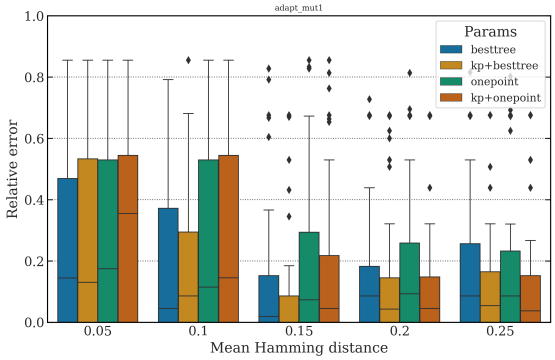
Figures 5.3a and 5.3b, the most significant difference between methods under the equal crossover probability is achieved around the middle of the mutation rates, i.e., from 0.3 to 0.7 at constant mutation and from 0.1 to 0.15 adaptive mutation. In these areas, the besttree method gains slightly over the one-point crossover. The results indicate that under higher mutation rates the construction potential of crossover operators is minimized. When considering the Crossover probability (cxpb), results depicted in Figures 5.4a and 5.4b clearly show that both methods tend to deliver better results under lower probability values.

From the performance perspective, the besttree method achieves better P_i values in most cases, between 30% and 100% higher gains, in comparison to the one point mate. This effect becomes substantial with intensifying mutation rates.

If we compare results at all configuration combinations in total at 900 instances per crossover operator, then the besttree achieved 33% ratio of optimal solutions found while the one point crossover achieved 27 %. At the best configurations, the besttree achieved 51 % and the one point crossover 48 % with no additional enhancement using constant mutation, 200 generations and 200 individuals.



(a) Relative error for both crossover methods under constant mutation (scenario const_mut1). The best results are achieved using the besttree method with 0.7 mutation probability.



(b) Relative error for both crossover methods under adaptive mutation (scenario adapt_mut1). The best results are achieved using the besttree method with 0.15 mean population Hamming distance limit.

Figure 5.3: Effect of crossover methods on relative error of evaluated instances.

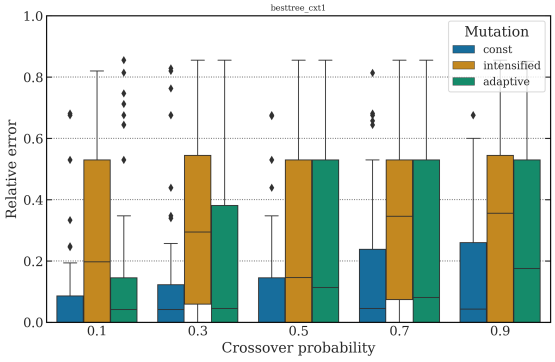
Mutation

As in the case of crossover, the first version of the GA was implemented using only a simple way of controlling the mutation level during the evolution process. Although the fixed-probability mutation was able to deliver an acceptable amount of optimal solutions, there was space for further improvements that could help to preserve or increase population diversity during the evolution. As described in 5.1.2, another two methods to control

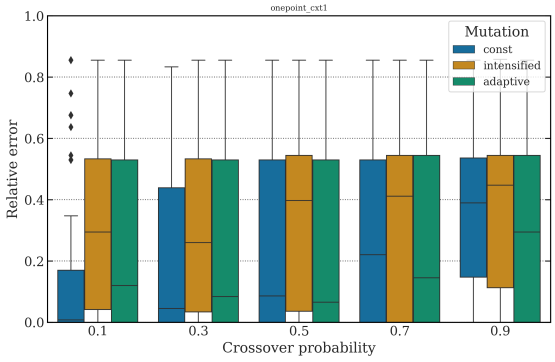
mutation rate were introduced: adaptive mutation with feedback and intensified mutation. Both mutation methods change mutation probability during the evolution in contrast to the constant mutation. It is important to remember that all methods control mutation rate but rely on the very same operator mutating the multicast tree.

From the comparison in Figures 5.4a and 5.4b, it is clear that the intensified mutation delivers significantly worse results than the two remaining methods. This tendency is most likely due to the fact, that the high mutation rate by the end of the evolution process may increase overall population diversity but does not give enough time to spread new genes or to cultivate some improving solution.

Globally at 600 instances per the mutation type, the constant mutation achieved a 44% ratio of optimal solutions, the adaptive 37 % and the intensified only 24 %. Considering only the best configuration combination containing 60 instances, the order remains unchanged with the constant mutation reaching $Opt = 60\%$, adaptive 46 % and intensified 33 %. Results are for scenarios with no additional enhancement operators using the besttree crossover, 200 generations and 200 individuals.



(a) Relative error for all mutation methods under different crossover mutations (scenario betstree_cxt1). The best results are achieved by constant mutation with $cxpb = 0.1$.



(b) Relative error for all mutation methods under different crossover mutations (scenario onepoint_cxt1). The best results are achieved by constant mutation with $cxpb = 0.1$.

Figure 5.4: Comparison of mutation methods under different crossover conditions.

Results depicted in 5.4a show that constant mutation is exceptionally successful in combination with the besttree crossover. This finding is at the same time validated in Figure 5.2a in a different scenario. In both result sets, the median relative error is equal to 0. In comparison to the adaptive mutation, the constant mutation is less sensitive to the change of mutation probability. A different situation can be observed in Figure 5.4b where the adaptive mutation reaches better results as the crossover probability increases. Although adaptive and constant mutations are controlled by different parameters, Figures 5.4a and 5.4b show the correlation between mutation probability and mean Hamming distance that consequently oscillates around some mutation probability. The trend is

evident, for example, at Mutation probability (mutpb) equal to 0.7 and Mean population Hamming distance limit (hdl) equal to 0.15.

The same finding as in the case of solution quality can be observed in the performance area. Although the values of P_i for constant mutation are slightly higher, performance results obtained for both methods are overlapping.

Taking a look at the detail of the constant mutation shown in Figure 5.5, one can see that the proposed GA can discover an optimal solution even under 40 generations and is able to break away from the local minimum repeatedly. This ability is mainly due to the high mutation rate that preserves population diversity leaving only a small space for the construction potential of the crossover operator. To leave the local minimum is more difficult as the algorithm gets closer closer to optimum since the number of improving solutions is usually shrinks. This explains the significant jumps in the fitness of the best individual in the population shown in the top left plot in Figure 5.5. The small improvements at the beginning are followed by stale areas without any improvement sometimes interrupted by an unexpected spike. Although the population can converge into a promising solution, the optimum may be entirely solitary in the search space, and so, only the high mutation rate may deliver entirely new building blocks.

The plot 5.5 shows that mean Hamming distance between the original and mutated individual is under a given configuration fluctuating between 0.2 a 0.25, effectively showing that one quarter of *alleles* change value during the mutation. The results also show that intensive mutation requires an increased application of the recovery algorithm. Last but not least, the plot showing the evolution of tree size confirms a finding from Figure 4.5 in Chapter 4 that the optimal solution for LDV tends to be based on large trees.

The adaptive mutation strives for the desired level of population diversity given by parameter hdl in contrast to the constant mutation that blindly keeps the mutation rate. The result of such control is shown in Figure 5.6 in the plot on the bottom right. The adapted mutation probability, at first, fluctuates to converge to the defined hdl, effectively leaving some time for the construction to be done by the crossover operator, but in the end, the probability intensifies to meet the desired hdl value. Such progress indicates that the preceding operators decrease population diversity and it is more challenging to keep the desired population diversity. Nevertheless, the fitness of the best individual in the population shows that even the adaptive mutation can get away from the local minimum after a stale period in the evolution. It should be noted that in both cases of detailed evaluation a local improvement operator was utilized.

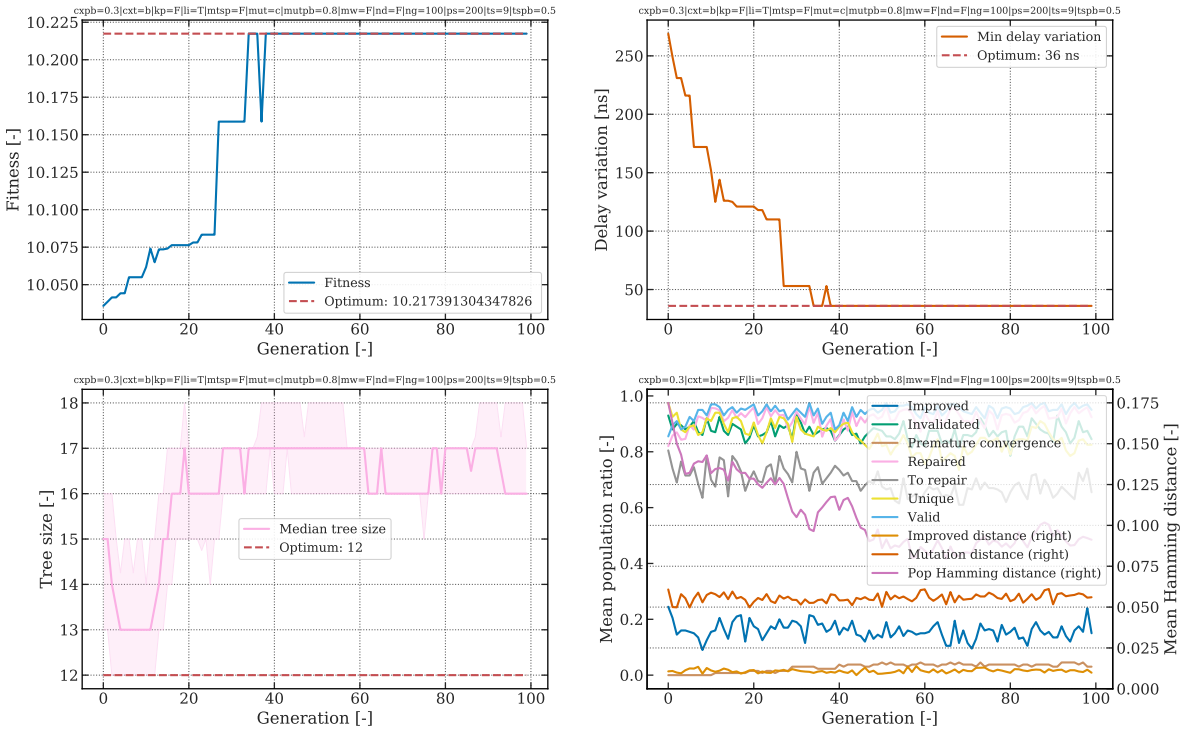


Figure 5.5: Detailed evolution process for constant mutation with probability 0.8 at an instance composed of 131 links and 25 nodes with 30% multicast coverage.

5.2.3 Enhancing operators

The proposed GA implements other potentially enhancing methods that can be optionally incorporated into the evolution process. However, methods are not always compatible with each operator. For example, No-Genotype Duplicates (nd) can hardly be beneficial when using adaptive mutation, as the control mechanism has only a little chance to keep the desired level of population diversity.

Local improvement

The first and possibly the most promising is the locally improving method that tries to implant enhancing alleles to the individual's gene using an objective-aware mutation, simulating the learning process. In case the individual is improved, it is passed to the next generation.

Degree of improvement is dependent on the type of mutation operator and mutation intensity. This phenomenon is depicted in Figure 5.7b, where Local improvement (li) obtains better results in terms of relative error, especially at $mutpb = 0.8$. A slightly different situation is observable in Figure 5.7a, where improvement methods find their use only in approximately half of the configurations. The overall performance given by P_i drops by 25% irrespective of the applied configuration that is caused mainly by the

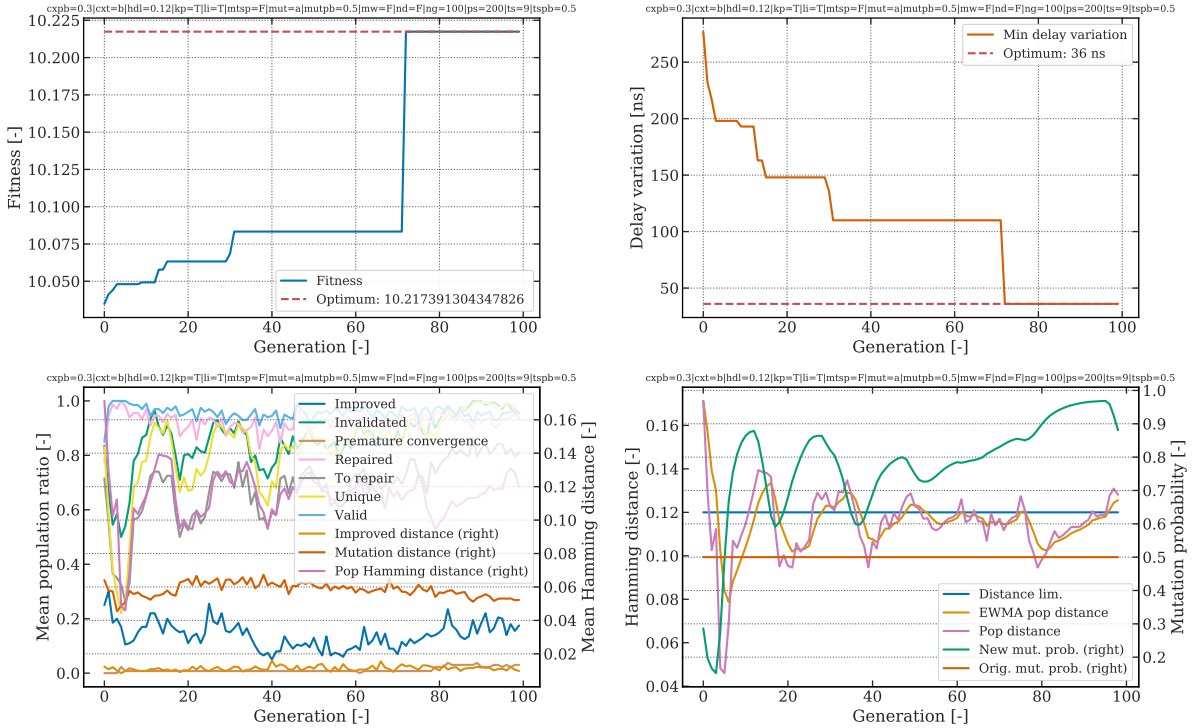
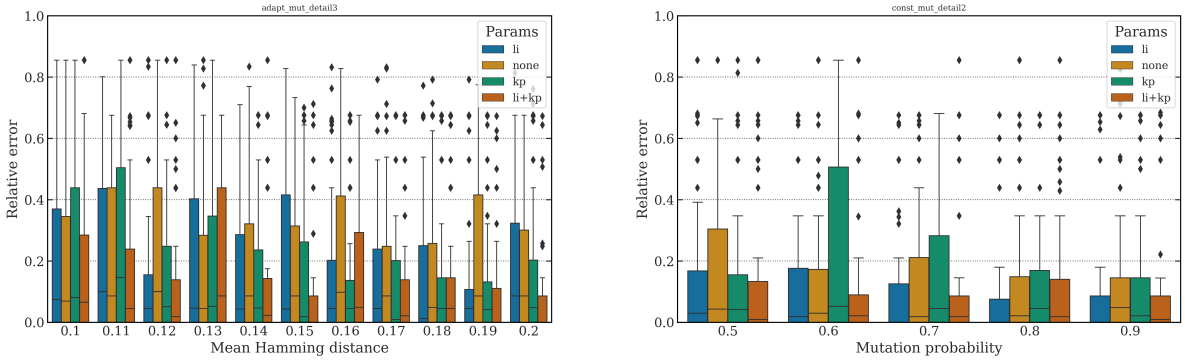


Figure 5.6: Detailed evolution process for adaptive mutation with hdl 0.12 at an instance composed of 131 links and 25 nodes with 30% muticast coverage.

extended computation time. Interestingly, one can notice a significant synergy between the li method and the elitism method named Keep best individual (kp). The synergy led to a 90% ratio, the highest of optimal results found among all scenarios (400 generations and 400 individuals).



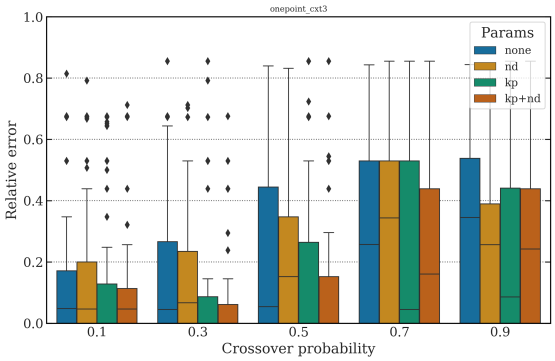
(a) Relative error achieved using the local improvement method under different adaptive mutation setups (scenario adapt_mut_detail3). The most improved part is in the area around $hdl = 0.18$.

(b) Relative error achieved using the local improvement method under different constant mutation probabilities (scenario const_mut_detail2). The best configuration is at $mutpb = 0.8$.

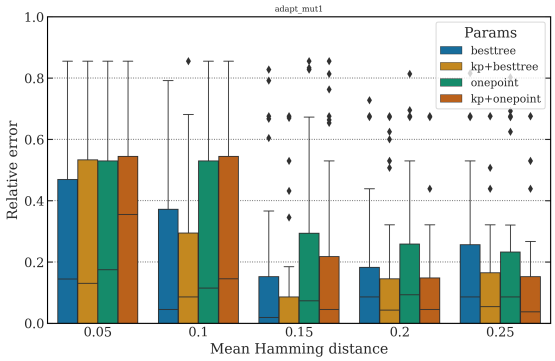
Figure 5.7: Comparison of the local improvement method with other potentially enhancing methods.

Elitism

Elitism, the method when the best individual in the population is preserved and passed to the next generation, demonstrated the potential to slightly improve results, especially for some specific configurations. Such an improvement is shown in 5.8b at both crossover operators for $hdl = 0.15$, and at $cxpb = 0.3$ in Figure 5.8a. It also acts in synergistic effect not only with the li operator but with the nd operator as well. From the performance perspective, the application of elitism is negligible.



(a) Relative error for configurations employing the kp method under different crossover probabilities (scenario onepoint_cxt3). The synergy between the kp and nd operator is strongest at $cxpb = 0.5$.



(b) Relative error in combination of the kp method and various crossovers, and mutation rates (scenario adapt_mut1). The positive impact of the kp is noticeable under intensive mutation.

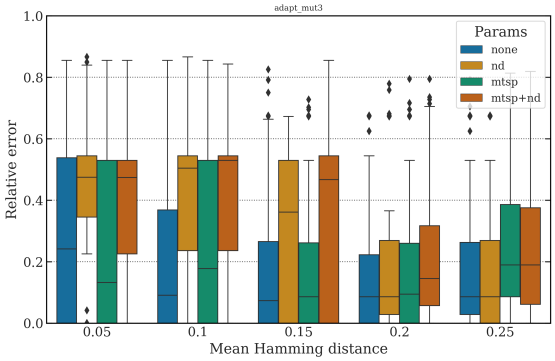
Figure 5.8: Effect of the kp method on quality of achieved results.

No-Genotype Duplicates

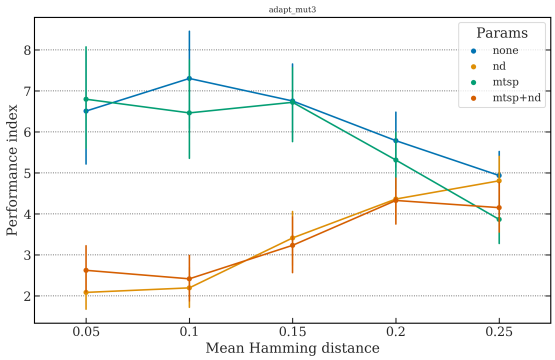
The nd operator used separately, without other enhancing operators, performed mostly poorly as is clear from the results in Figure 5.9a. The not very promising quality of results is, in addition, degraded by low P_i values depicted in 5.9b. This degradation is mainly caused by the repeatedly invoked creation of new individuals that is time-consuming. The nd operator is incompatible with the adaptive operator as it prevents the control mechanism to keep the desired population-diversity level. The convergence of results at the end of the plot is caused by a high mutation rate when almost all individuals in the population are affected by both operators.

Objective-aware Mutation

The last evaluated operator is the enhanced mutation operator, i.e., not a mechanism controlling the mutation rate. The li operator inspires this operator as it. It consciously chooses branches of the multicast tree in the order given by the objective function as



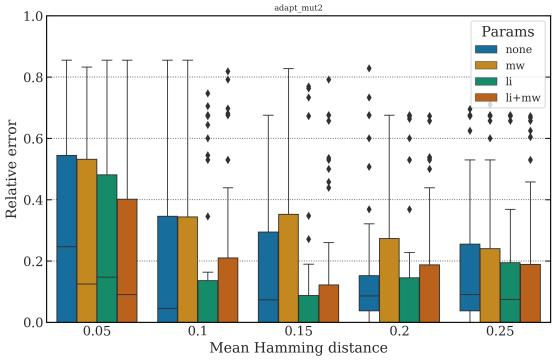
(a) Relative error for adaptive mutation with the nd operator under different mutation rates (scenario adapt_mut3). The incompatibly is obvious at lower mutation rates.



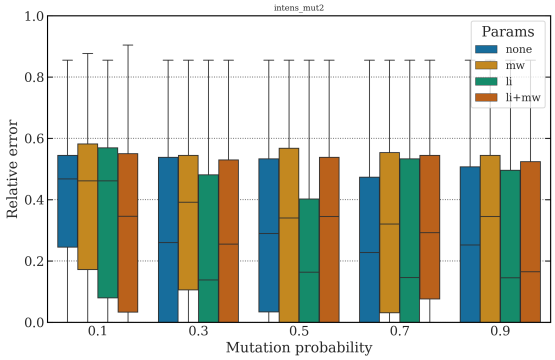
(b) Resulting P_i results in Figure 5.9a. When the mutation rate is saturated the difference in performance vanishes.

Figure 5.9: Impact of the no-genotype duplicates operator on the adaptive mutation.

described in Section 5.1.2. Evaluation results in Figures 5.10a and 5.10b clearly show that this approach is not beneficial. Almost none of the explored configurations lead to improving solutions in comparison to configurations not employing the enhanced mutation operator. In this case, the requirement for variation operator to not bias the population stated in [159] is correct.



(a) Relative error for adaptive mutation with the Objective-aware mutation (mw) operator under different mutation rates (scenario adapt_mut2). A slight improvement at lower mutation rate is negligible in compare to other configurations.



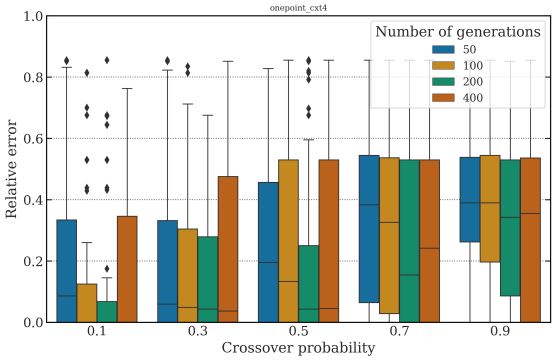
(b) Relative error for intensified mutation with the mw operator under different mutation rates (scenario intens_mut2). In combination with intensified mutation the mutation operator deteriorate the solution quality.

Figure 5.10: Effect of objective-aware mutation on the quality of solutions.

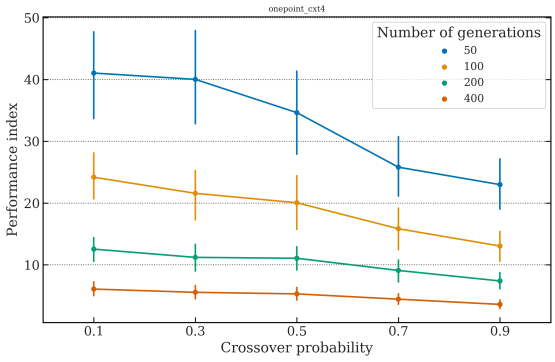
5.2.4 Other parameters

Number of generations

The number of generations the evolution process iterates through has a positive impact on the solution quality but only to some limit. After then the relative error does not improve, or it may deteriorate as one can see in Figure 5.11a. The turning point seems to be 200 generations. This finding conforms with the latest evolutionary findings suggesting the use of fewer generations [166]. The performance impact is intuitive as shown in Figure 5.11b. More generations mean longer evaluation times, but not necessarily excellent results.



(a) Relative error achieved for different number of generations under different crossover probabilities (scenario onepoint_cxt4). The turning point is at 200 generations.



(b) P_i for different number of generation under different crossover probabilities (scenario onepoint_cxt4). The higher the number of generations the lower the P_i values.

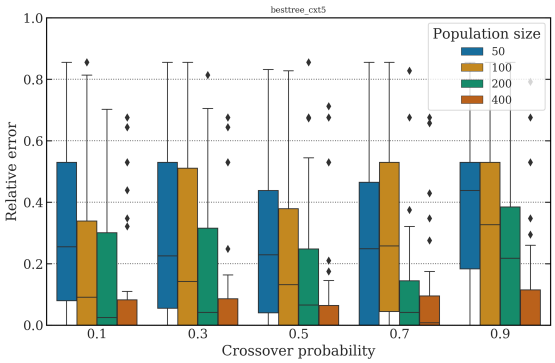
Figure 5.11: Impact of number of generations on the solution quality and performance of GA.

Population size

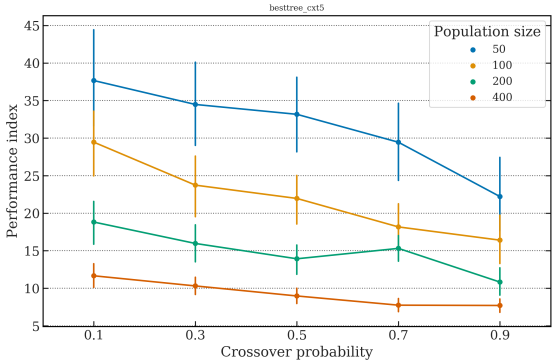
In contrast to the number of generations, the evaluation of population size delivers expected results. These can be summarized as: the larger the population, the lower the relative error, and the larger the population, the worse the GA performance. The progress in the drop of relative error is evident in all cases in Figure 5.12a, and the corresponding performance in 5.12b.

5.3 Hyperparameter optimization

While in the first section of evaluation the focus was put on the isolated effect of the implemented genetic operators in conjunction with different parameters, the second part of the evaluation aims to investigate what is the best combination of operators and respective parameters. Although it may seem that evolutionary algorithms have been known for a



(a) Relative error achieved for various population sizes under different crossover probabilities (scenario besttree_cxt5). The best results are achieved with 400 individuals at $cxpb = 0.5$.



(b) P_i for various population sizes under different crossover probabilities (scenario besttree_cxt5). The larger the population, the worse the performance, but the higher the probability of reaching the optimal solution.

Figure 5.12: Impact of population size on the solution quality and performance of GA.

long time, and so, one would expect some generally suitable configuration in the parameter space to exist, the opposite is true.

At this point, it is necessary to distinguish between between parameters and hyperparameters. While parameters evolve during the model execution the hyperparameters are configured before the model is executed. Therefore, the latter ones are the model's input parameters. The term hyperparameters is more common in the area of machine learning.

This topic of optimal hyperparameters for GA remains unclear, as show authors in [167]. This recently published paper contains one of the most extensive explorations of parameter space on a broad range of problems. Although the authors make some interesting observations, they conclude the work with the comment: "the only trend that seems to emerge is there being very little trend." As was mentioned at the beginning of this chapter, the area of evolutionary algorithms and the range of implementation is so widespread that there is little chance to find some universal set of hyperparameters.

Since the proper model configuration is crucial for reaching high-quality results, the following section sets a goal to find the best hyperparameters and operator variants for the proposed model.

5.3.1 Scenarios

As the number of configuration combinations became huge, grid optimization was nearly impossible. To tackle this issue, the first idea was to explore only such values of hyperparameters and configuration combinations that seemed to be promising. The number of scenarios already computed in the previous part of the evaluation was relatively large, so these results were reused to find the best configuration.

The overall results of the 20 best scenarios and related hyperparameters are listed in Table 5.1. The scenarios are sorted in descending order by the ratio of optimal solutions reached using the declared hyperparameters. At first sight, it is evident that scenarios with the fixed-probability mutation significantly overtook the rest of the field as they occupied leading positions.

However, another evaluation criterion indicates that the most beneficial set of hyperparameters for practical use is not the best one. It is not surprising that the configuration with a large population and number of generations occupies the first position. The interesting performance results lie on the third and fourth rows, using only 100 generations. Although these scenarios reach only slightly better results in the context of the optimal solutions, they excel both in performance index p_i and speedup S_l. The achieved values of these metrics are beyond the surrounding results. Following this perspective and in the context of the limited number of scenarios, the best option of configuration and hyperparameters for the proposed GA comes from the *const_ mut_ detail2* scenario. By nature of scenarios and the limited number of combinations, we cannot disprove that there exist other more efficient configurations, but for our purposes this evaluation is sufficient.

5.3.2 Bayesian optimization

Since the parameter search space is enormous, the only reasonable way to seek the best hyperparameters is to employ some method that crawls the space in an informed manner. A powerful Bayesian optimization method was chosen to address this problem as the method is well researched and widely adopted for optimizing computationally expensive objective functions [168].

At a glance, the method builds a probability model, termed as surrogate function, of the objective function $f : \chi \to \mathbb{R}$ and make use of the model to find promising hyperparameters to be evaluated in the original objective function. At our optimization task, the hyperparameter optimization can be expressed by an equation in (5.14), where $f(x)$ is the objective function to be minimized and $x^\star$ is a set of hyperparameters from the parameter domain χ that delivers the lowest objective value.

$$x^\star = \underset{x \in \chi}{argmin} f(x) \tag{5.14}$$

The objective function $f(x)$, known as loss function, does need to be the original function coming from the initial problem, but it may express other relevant metrics as the average relative error of evaluated instances. In case of the LDV problem, we kept the original objective function with an only a slight modification of summing the achieved

delay deltas across all instances from a set of instances I as shown in (5.15).

$$f(x) = \sum_{i \in I} \delta_{T_i} \tag{5.15}$$

As the authors of [169] note, methods based on Bayesian optimization are effective in practice even if the underlying f being optimized is stochastic, non-convex, or even non-continues. The nature of the LDV problem and the proposed GA implementation perfectly suits this method as the optimal solution can be very spiky in the search space. Moreover, the method may be used not only to find promising hyperparameters but a combination of enhancing operators to employ as well.

Bayesian optimization may be used in various models. These models profit from the Bayesian reasoning based on the historical track of the objective function evaluations when the method iterates over promising hyperparameters and gradually updates the surrogate of the objective function. The model we employed is known as Sequential Model-Based Optimization (SMBO), at which an inner loop of the algorithm runs trials sequentially applying the potentially better hyperparameters.

The SMBO model operates within a search domain with some objective function, surrogate model and a selection function. The search domain comes from the GA, and the objective function was defined in (5.15). The selection function is used to obtain the next set of hyperparameters from the surrogate model. The commonly used selection criteria incorporate an Expected Improvement (EI) approach [170] where the goal is to maximize EI concerning the given hyperparameter. The last piece is the surrogate model of the objective function, but as always there exist numerous approaches such as Random Forest Regressions, Gaussian Processes, Tree Parzen Estimators (TPE). Thankfully, results published in [171] point to TPE as it can deliver the lowest validation error in the shortest time. Limited by the number of trials we can evaluate in a reasonable time, the TPE surrogate model was the preferred one.

Describing all parts of the optimization process and detailing particular steps is out of the scope of this thesis, therefore I refer the reader to [168], [169] for further reading.

Evaluation results

The optimization approach described above would be time-consuming to implement, and so, a proper framework called Hyperopt [172] was chosen for this task. The framework was written as a Python module and it supports both the SMBO and TPE models which makes the hyperparameter search implementation very convenient.

As in the case of the GA evaluation the set instances remain identical, only the number of iterations per instance was decreased to 1. At first, the surrogate is built by observation

of the objective function results, and secondly, the Bayesian optimization accepts that the observation process is subject to noise. Since the optimization process naturally converged to large population sizes, the reduction of evaluations per instance allowed the process to run for more steps in a reasonable time. The available computation window was 7 days long for each hyperparameter domain allowing 500 iterations. Seven researched domains were defined and are described in Appendix A.2, with the goal to cover all reasonable configuration combinations and parameter search space.

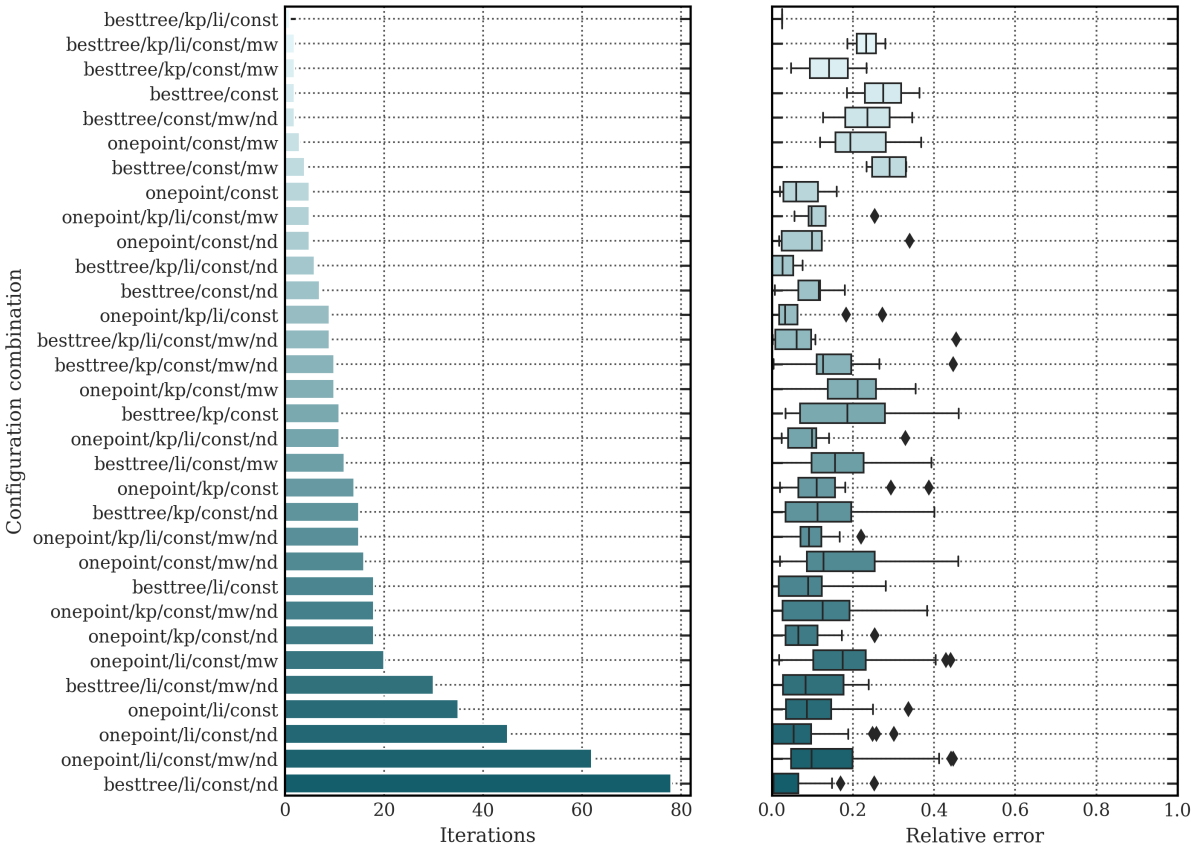


Figure 5.13: Searched binary configuration combinations and related relative errors for the *tpe_ const_ mut* domain using the Bayesian optimization algorithm. The combination of besttree crossover, local improvement, and no-genotype duplicates operators deliver the most promising results.

The optimization results are summarized in Table 5.2. The first observation is that hyperparameters were found in all domains leading to a global optimum at all instances from the evaluation set. The ratio of setups where all instances reached an optimal solution is noted as *Opt*. Unequivocally, the best results were yielded with the constant mutation in the *tpe_ const_ mut* domain. In this domain, 10 % of all evaluations found an optimal solution. Taking into account the successful configurations at each domain, the *tpe_ const_ mut* performs better in both the performance index p_i and speedup S_l as well.

The finally reported combination of parameters may be slightly misleading because the share of particular GA features may be irregular in the optimal parameter configurations. We need to take a look on a cruise of the optimization algorithm in the parameter search space to understand better what is behind the 10 % of optimal configurations.

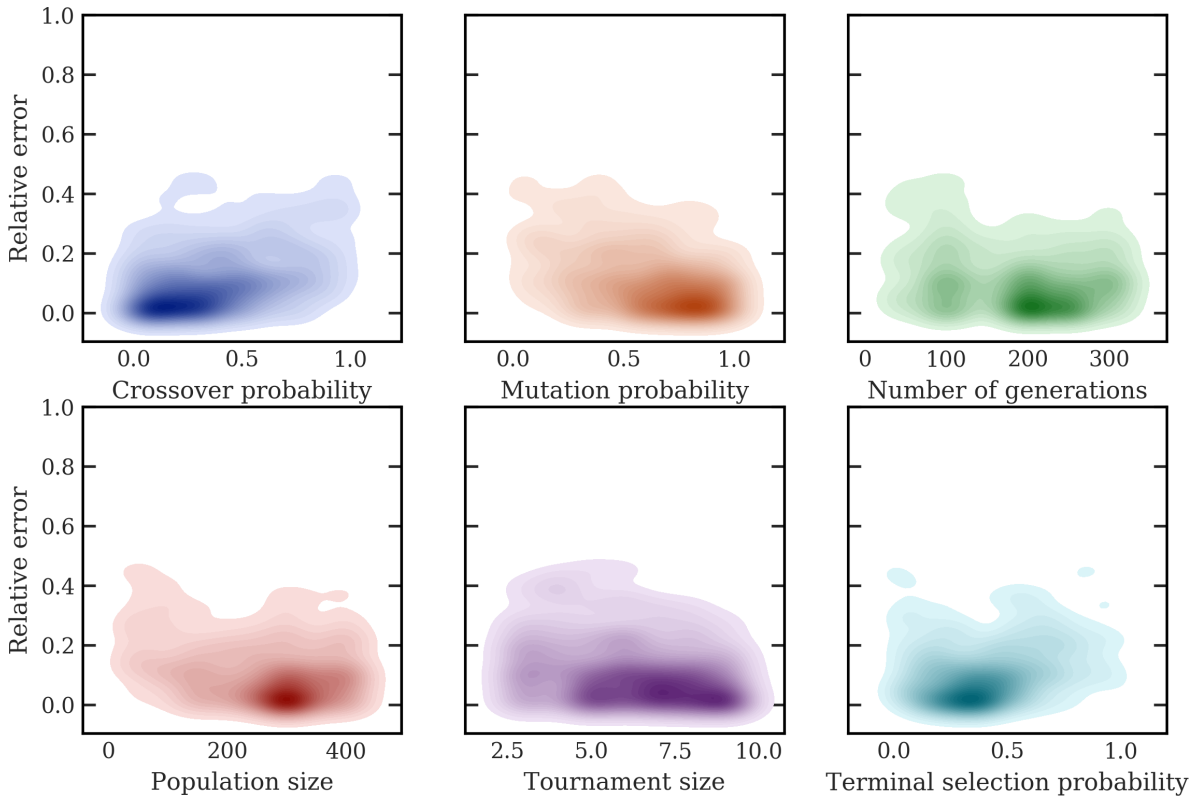


Figure 5.14: Searched numerical hyperparameters in relation to relative errors for the *tpe_const_mut* domain using the Bayesian optimization algorithm. Dark areas show where the algorithm concentrated the search for the best parameters.

The parameter search space can be divided into hyperparameters acquiring values from a numerical range (probabilities, sizes), and methods that are employed in a particular configuration combination. Such GA's methods can be activated or deactivated and, thus, in the context of hyperparameter optimization are of the binary set.

Looking for the best configuration, it is convenient to examine the relation between the combination of configuration values and the relative error achieved under such a setup. The combination of plots in Figure 5.13 clearly shows the most promising areas in the binary parameter search space where the optimization algorithm concentrated its effort. The surrogate function reported the best results around the configuration combination of the besttree crossover, local improvement, and no-genotype duplicates operators in conjunction with the constant mutation. As one can see, the median of relative error is equal to zero with a very narrow Interquartile range (IQR) which means that most of the evaluation in this area was very successful. The configuration, most likely, reported from

the last optimal trial is placed on the fourth place.

From the perspective of numerical hyperparameters, there are no surprises in comparison to binary ones. To illustrate the behavior of the optimization algorithm, the numerical parameters in relation to relative error are plotted in Figure 5.14 using a bivariate kernel density estimation. The interpretation is, in our case, straightforward: the darker the plotting the higher the intensity of the algorithm devoted to the given parameter value. In comparison to results in Table 5.2, hyperparameter values are correspond to areas in kernels.

5.4 Evaluation summary

Although the proposed GA is complex, the evaluation results show that it can deliver valid, feasible, and often optimal, solutions even for large instances. In comparison to the ILP model introduced in the previous chapter, the implemented GA proved itself to be highly performant as the reported speed up in many cases exceeds lower hundreds, especially at larger instances. Moreover, this is valid under the high ratio of optimal solutions reaching 90 %. While the proposed crossover and mutation operators passed the evaluation solidly, the methods for the control of population diversity did not work convincingly. Surprisingly, the best results were persistently reached by the fixed-probability mutation. On the contrary, the local improvement method is satisfactory as it was in many cases selected as a part of the best configurations.

The hyperparameter optimization in both approaches, using scenarios and Bayesian optimization, led to very similar results, though slightly different at some numerical parameters. It turns out that the most important is a high mutation rate fluctuating around 0.75 in combination with a relatively low crossover probability at 0.3. The high level of disruption required to find an optimal individual denies Holland's construction theory which is probably due to the spiky nature of optimal solutions in the case of the LDV problem.

From the implementation perspective, to design, debug, and evaluate the proposed GA took many months and in the end thousands of CPU hours.

Table 5.1: Best configurations and hyper parameters obtained for top 20 scenarios, sorted by ratio of optimal solutions *Opt*. The grayed rows indicates configurations with advantageous results in terms of performance index p_i and speed up S_l.

Scenario	*Opt*	ϵ	p_i	S_l	cxt	cxpb	hdl	kp	li	mut	mutpb	mw	ng	ps	ts
const_mut_detail1	0,85	0,05 ± 0,17	3,37 ± 1,91	40,27 ± 69,67	besttree	0,3		0	0	const	0,8	0	400	400	9
const_mut1	0,72	0,08 ± 0,18	8,62 ± 3,69	62,56 ± 98,59	besttree	0,5		1	0	const	0,7	0	200	200	4
const_mut_detail2	0,66	0,09 ± 0,19	17,09 ± 8,27	219,15 ± 323,26	besttree	0,3		0	1	const	0,8	0	100	200	9
adapt_mut_detail3	0,62	0,08 ± 0,16	14,91 ± 6,16	175,15 ± 250,83	besttree	0,3	0,20	1	1	adapt	0,5	0	100	200	9
besttree_cxt5	0,63	0,09 ± 0,20	10,31 ± 4,82	103,50 ± 170,69	besttree	0,3		0	0	const	0,5	0	100	400	4
const_mut6	0,62	0,11 ± 0,21	8,94 ± 4,21	101,70 ± 168,15	besttree	0,5		0	0	const	0,7	0	200	200	9
besttree_cxt2	0,67	0,09 ± 0,18	8,27 ± 4,12	90,14 ± 155,50	besttree	0,3		0	1	const	0,5	0	200	200	4
besttree_cxt3	0,63	0,11 ± 0,21	7,81 ± 3,85	65,12 ± 102,73	besttree	0,9		1	0	const	0,5	0	200	200	4
const_mut2	0,64	0,11 ± 0,20	7,01 ± 3,28	64,34 ± 84,10	besttree	0,5		0	0	const	0,9	1	200	200	4
adapt_mut2	0,65	0,12 ± 0,22	6,79 ± 3,12	62,44 ± 95,45	besttree	0,5	0,15	0	1	adapt	0,5	1	200	200	4
const_mut5	0,62	0,12 ± 0,22	6,65 ± 3,52	53,55 ± 64,14	besttree	0,5		0	0	const	0,7	0	100	400	4
const_mut4	0,62	0,13 ± 0,23	6,10 ± 3,18	49,79 ± 64,11	besttree	0,5		0	0	const	0,7	0	200	200	4
besttree_cxt4	0,62	0,10 ± 0,22	5,04 ± 2,72	62,23 ± 94,88	besttree	0,3		0	0	const	0,5	0	400	200	4
adapt_mut_detail1	0,63	0,11 ± 0,22	3,07 ± 2,00	41,24 ± 72,30	besttree	0,3	0,10	0	0	adapt	0,5	0	400	400	9
onepoint_cxt4	0,52	0,13 ± 0,23	24,18 ± 14,83	346,48 ± 611,59	onepoint	0,1		0	0	const	0,5	0	100	200	4
onepoint_cxt3	0,53	0,12 ± 0,22	17,78 ± 11,02	245,96 ± 433,98	onepoint	0,3		1	0	const	0,5	0	200	200	4
onepoint_cxt2	0,58	0,12 ± 0,21	17,71 ± 11,19	247,53 ± 436,63	onepoint	0,1		0	0	const	0,5	0	200	200	4
adapt_mut_detail2	0,58	0,15 ± 0,25	16,06 ± 9,93	223,39 ± 383,53	besttree	0,3	0,12	1	1	adapt	0,5	0	100	200	9
besttree_cxt6	0,52	0,13 ± 0,23	11,89 ± 7,67	167,67 ± 293,10	besttree	0,1		0	0	const	0,5	0	200	200	7
besttree_cxt1	0,60	0,10 ± 0,19	10,43 ± 5,42	125,48 ± 212,77	besttree	0,1		0	0	const	0,5	0	200	200	4

Table 5.2: Best configuration parameters as reported by the Bayesian optimization algorithm in defined parameter search domains.

Domain	Opt	p_i	S_l	cxt	cxpb	hdl	kp	li	mtsp	mut	mutpb	mw	nd	ng	ps	ts	tspb
tpe_const_mut	0,10	10,19 ± 6,02	145,77 ± 277,07	onepoint	0,33		0	1		const	0,75	0	0	200	250	7	0,42
tpe_onepoint_cxt	0,06	7,70 ± 4,25	107,59 ± 204,66	onepoint	0,47		1	1		const	0,92	0	0	200	400	9	0,35
tpe_adapt_mut	0,03	7,59 ± 4,21	106,48 ± 203,05	onepoint	0,95	0,24	1	0		adapt	0,38	0	0	300	350	8	0,32
tpe_onepoint_cxt2	0,02	5,86 ± 3,46	85,46 ± 166,17	onepoint	0,07	0,19	0	0		adapt	0,34	1	0	250	350	4	0,34
tpe_besttree_cxt	0,08	5,84 ± 3,12	80,12 ± 150,99	besttree	0,78		0	1		const	0,59	0	1	150	300	6	0,15
tpe_adapt_mut2	0,01	4,67 ± 1,77	52,12 ± 90,23	besttree	0,73	0,16	1	1	1	adapt	0,63	0	0	250	300	5	
tpe_besttree_cxt2	0,02	5,43 ± 1,53	48,13 ± 77,25	besttree	0,46	0,14	1	0		adapt	0,71	0	0	300	350	3	0,45

If knowledge can create problems, it is not through ignorance that we can solve them.

Isaac Asimov

6

Multi-tree BDLDV Multicast Problem

Even though the three previous chapters give the reader of this thesis the impression of thematic isolation, the following chapter links the observations and results into a comprehensive model of the multi-tree BDLDV multicast problem. Extending the original LDV problem simultaneously by a bandwidth constraint shared by multiple multicast groups, the problem is even more demanding.

Since most of the mathematical and algorithmic instruments were extensively detailed in related chapters, the following sections aim mostly to describe models and their evaluation. At first, the chapter starts with a compact ILP model built on the formulation of the LDV model presented in Section 4.2.1. The scalability of the ILP solution is limited, and due to this reason, a hybrid decomposed model is introduced in conjunction with the evolutionary approach.

The chapter is concluded with an evaluation section that steps out of the numerical evaluation to extensive simulations to verify the original application idea. Is it possible to successfully transfer the computed multi-tree BDLDV forwarding employing the SDN concept to the environment of non-deterministic Ethernet networks, and if so, how does it perform in comparison to MST or SPT? Custom simulation models were implemented, incorporating values of switch latency obtained in Chapter 3, to answer this question.

6.1 Proposed ILP models

Since the multi-tree BDLDV problem is significantly complex in comparison to the LDV problem presented in Section 4.2.1, the underlying network model and formal problem definition have to be refined.

6.1.1 BDLDV problem definition

Network model

Firstly, let us remind the network model and extend it to the bandwidth component. The network is modeled as a directed connected graph $G = (V, L)$ where V is a set of network nodes, and L is a set of network links. The set of nodes V represents inter-connecting nodes, e.g., Ethernet switches. The publisher and subscribers are connected to these nodes.

All links are bidirectional, each directed link $\ell = (u, v), \ell \in L$ going from $u \in V$ to $v \in V$ has a counterpart $\ell' = (v, u)$ in the opposite direction from $v \in V$ to $u \in V$. Each node $v \in V$ is incident to a set of ingress links $\omega^+(v)$ and egress links $\omega^-(v)$. Real non-negative value is assigned to every link $\ell \in L$ in the form of a link delay $d_\ell \to R^+$. The link delay function d_ℓ is a measure of link propagation delay. The function is naturally symmetrical, therefore $d_\ell = d_{\ell'}, \ell \in L, \ell' \in L$. Similarly, each node $v \in V$ has assigned a real non-negative function value $d_v \to R^+$ describing node switching latency. The delay caused by the queuing effect is neglected.

Moreover, each link $\ell \in L$ has been assigned a real non-negative link bandwidth function $B(\ell) \to R^+$. Bandwidth function $B(\ell)$ is a residual bandwidth available on the link $\ell \in L$. The initial residual bandwidth, i.e., when no traffic occupies the link, is equal to link capacity c_ℓ. Since the traffic load in the network is not spread equally on the links, link functions are asymmetrical. Therefore, it often occurs that $B(\ell) \neq B(\ell'), \ell \in L, \ell' \in L$.

A single multicast tree $T(v_p, S)$ is a sub-graph of G compounded of a multicast source node (publisher) $v_p \in V$, and multicast destination nodes (subscribers) $S \subseteq V \backslash \{v_p\}$ where the set $S \cup \{v_p\}$ is called a multicast group. Set S and the publisher node v_p are interconnected by links through a subset of Steiner nodes $I \subset V$ which form a part of $T(v_p, S)$.

Problem definition

As the network model is clear, the multi-tree BDLDV multicast problem can be formulated. Let $P_T(v_p, v_s), v_s \in S$ be a set of links $\ell \in L$ on a path from node v_p to node v_s in the tree $T(v_p, S)$, and $I_T(v_p, s) \subset V$ a set of nodes along this particular path including

both nodes v_p and v_s. The total end-to-end transmission delay $D_T(v_p, v_s)$ is then a sum of all link and node delays along the path as given in expression (6.1).

$$D_T(v_p, v_s) = \sum_{\ell \in P_T(v_p, v_s)} d_\ell + \sum_{v \in I_T(v_p, v_s)} d_v \tag{6.1}$$

The bandwidth bottleneck of the path $P_T(v_p, v_s)$ is expressed as the minimum residual bandwidth available along the path from v_p to v_s in equation (6.2).

$$B_T(v_p, v_s) = min\{B(\ell) | \forall \ell \in P_T(v_p, v_s)\} \tag{6.2}$$

The delay-variation δ_T of the multicast tree $T(v_p, S)$ is defined as a maximum difference among end-to-end delays along paths of all node pairs $\mathcal{PS} = \{\{v_p, v_s\} | \forall v_s \in S\}$ as described by expression (6.3).

$$\delta_T(v_p, S) = max\{|D_T(v_p, u) - D_T(v_p, v)| | \forall u, v \in S\} \tag{6.3}$$

Let Δ be the end-to-end delay constraint and β the bandwidth constraint for all node pairs in $\mathcal{PS}$. The BDLDV multicast problem is defined as a minimization of $\delta_T(T(v_p, S))$ subject to constraints expressed in (6.4) and (6.5).

$$D_T(v_p, v_s) \leq \Delta \qquad \forall (v_p, v_s) \in \mathcal{PS} \tag{6.4}$$

$$B_T(v_p, v_s) \geq \beta \qquad \forall (v_p, v_s) \in \mathcal{PS} \tag{6.5}$$

Multi-tree extension

The problem definition given above for a single multicast tree is valid for the multi-tree problem as well. The multi-tree multicast problem extends the original problem by a set of vectors $\Gamma = \{M_0 ... M_{g-1}\}$ for g multicast groups, where each vector describes the particular group by a quartet of variables according to $M = \langle v_p, S, \beta, \Delta \rangle$. While the sense of each of the variables in the M vector corresponds with the description provided in the previous text, in the following models, these variables are accompanied by superscript M to emphasize its membership to a particular multicast group.

6.1.2 Multiple Shortest Path Trees model

As in the case of the LDV problem, the trivial reference for the multi-tree BDLDV problem is based on SPT, as defined in Section 4.2.1. However, the formulation has to be adapted to the condition of multiple multicast groups; thus, Multiple Shortest Path Trees (MSPT)

is introduced. It should be noted that this formulation neither reflects link capacity nor transmission delays, as expected.

The goal of the objective function in (6.6) is to minimize total tree size per multicast group $M \in \Gamma$, where $y_\ell^M \in \{0,1\}$ with $y_\ell^M = 1$ if traffic from v_p^M to $v_s^M \in S^M$ is forwarded on link ℓ, or $y_\ell^M = 0$ otherwise.

$$\min \left(\sum_{M \in \Gamma} \sum_{\ell \in L} y_\ell^M \right) \tag{6.6}$$

Since the constraints remain similar to the constraints in Section 4.2.1, we only briefly recall the meaning of the equation blocks. The first one is the obligatory *flow conservation constraint* in (6.7), where $\varphi_\ell^{ps_M} \in \mathbb{R}^+$ denotes flow at link ℓ from v_p^M to destination $v_s \in S^M$. This condition must be satisfied for each pair of nodes in $\mathcal{PS}^M = \{\{v_p^M, v_s\} | \forall v_s \in S^M\}$.

$$\sum_{\ell \in \omega^+(v)} \varphi_\ell^{ps_M} - \sum_{\ell \in \omega^-(v)} \varphi_\ell^{ps_M} = \begin{cases} 1 & \text{if } v = v_p \\ -1 & \text{if } v = v_s \\ 0 & \text{otherwise} \end{cases} \qquad v \in V, (p,s)_M \in \mathcal{PS}^M, M \in \Gamma \tag{6.7}$$

The rest of the ILP formulation in (6.8)–(6.11) ensures that any solution found is a tree composed of all flows through binary vector y_ℓ.

$$\varphi_\ell^{ps_M} \leq y_\ell^M \qquad \ell \in L, (p,s)_M \in \mathcal{PS}^M, M \in \Gamma \tag{6.8}$$

$$y_\ell^M < \varphi_\ell^{ps_M} + 1 \qquad \ell \in L, (p,s)_M \in \mathcal{PS}^M, M \in \Gamma \tag{6.9}$$

$$\sum_{\ell \in \omega^+(v)} y_\ell^M \leq 1 \qquad v \in V, M \in \Gamma \tag{6.10}$$

$$\varphi_\ell^{ps_M} \geq 0 \qquad \ell \in L, (p,s)_M \in \mathcal{PS}^M, M \in \Gamma \tag{6.11}$$

6.1.3 Compact multi-tree BDLDV multicast model

Likewise, the MSPT is analogous to SPT, as the multi-tree BDLDV problem formulation shares most of its logic with the ILP model for the LDV problem. The substantial extensions are de facto only three. The first one is an adaptation of the objective function in (6.12) to support multiple multicast groups. As one can see, the objective is now minimized over the sum of all delay variations.

$$\min(\sum_{M\in\Gamma} \delta^M_{max} - \delta^M_{min}) \tag{6.12}$$

The second and third extensions of the original LDV comes out of the refined multi-tree BDLDV problem in Section 6.1.1. The constraint tightening of the delay variation defined in (6.13) is now comprised of the sum of link propagation delays and switch latencies. To capture the end-to-end delay correctly, we need to account for store-end-forward delay occurring at each hop. This requirement is ensured by the fraction $\frac{p^M_\lambda}{c_\ell}$, where p^M_λ is packet length used to generate flow associated to multicast group M (including header, preamble, SFD, and CRC), and bandwidth capacity $c_\ell \in \mathbb{R}^+, \ell \in L$. The maximum end-to-end delay is limited from above as defined in (6.14). The last additional constraint (6.15) ensures that any link $\ell \in L$ used to construct the multicast tree consuming bandwidth β^M will not exceed link capacity c_ℓ.

$$\delta^M_{min} \leq \sum_{\ell\in L} \left(\varphi_\ell^{v^M_p s} \left(d_\ell + \frac{p^M_\lambda}{c_\ell} + d_{(v=SRC(l))} \right) \right) + d_{(v=s)} \leq \delta^M_{max} \qquad \ell \in L, s \in S^M \tag{6.13}$$

$$\delta^M_{max} \leq \Delta^M \qquad M \in \Gamma \tag{6.14}$$

$$\sum_{M\in\Gamma} y^M_\ell \beta^M \leq c_\ell \qquad l \in L \tag{6.15}$$

Naturally, the flow conservation constraint expressed in (6.7) and tree constraints defined in (6.8)–(6.11) remain preserved for this model. In addition to the MSPT model, the modification of the objective function may cause the emergence of loops. Therefore, auxiliary constraints corresponding to those in Section 4.2.1 are introduced. The idea behind the constraints is similar. The primary constraint (6.16) assures that all links assigned to a particular flow $\varphi_{\ell_i}^{ps_M}$ are virtually labeled in non-decreasing order in vector o^{ps}_ℓ. The constraints are limited only to a set of adjacent pairs of ingress/egress links $\mathcal{IO} = \{\{\ell_i, \ell_o\} | \ell_i \in \omega^+(v), \ell_o \in \omega^-(v), v \in V\}$. Constraints (6.17)–(6.19) express that ℓ_i and ℓ_o are ingress and egress links along with a specific flow $(p, s)_M \in \mathcal{PS}^M$. Typically, this information can be obtained by logical operation AND for these decision variables. However, operation AND is a non-linear operation; therefore, the standard linearization

approach was applied.

$$o_{\ell_i}^{ps_M} - o_{\ell_o}^{ps_M} \geq a_{io}^{ps_M} \quad (i,o) \in \mathcal{IO}, \ell_i \in \omega^+(v), \ell_o \in \omega^-(v), v \in V, (p,s)_M \in \mathcal{PS}^M \tag{6.16}$$

$$\varphi_{\ell_i}^{ps_M} - b \geq a_i^{ps_M} \qquad \ell_i \in \omega^+(v), v \in V, (p,s)_M \in \mathcal{PS}^M, M \in \Gamma \tag{6.17}$$

$$\varphi_{\ell_o}^{ps_M} - b \geq a_o^{ps_M} \qquad \ell_o \in \omega^-(v), v \in V, (p,s)_M \in \mathcal{PS}^M, M \in \Gamma \tag{6.18}$$

$$a_i^{ps_M} + a_o^{ps_M} - 1 \leq a_{io}^{ps_M} \quad (i,o) \in \mathcal{IO}, (p,s)_M \in \mathcal{PS}^M, M \in \Gamma \tag{6.19}$$

$$a_{io}^{ps_M} \in \{0,1\} \qquad (i,o) \in \mathcal{IO}, (p,s)_M \in \mathcal{PS}^M, M \in \Gamma \tag{6.20}$$

$$\tag{6.21}$$

6.1.4 Decomposed multi-tree BDLDV multicast model

Although the compact model can reliably deliver valid solutions, it was clear from the onset that extending the LDV model to the BDLDV ILP formulation will suffer by scalability issues even more. Therefore, the compact model is used to reasonably explain the idea behind the optimization process and later during the evaluation as a reference model.

Reviewing the compact model, one can notice that adding the bandwidth constraint in (6.15) interconnects the separated LDV problems together. Neglecting the bandwidth constraint, the ILP problem falls apart to sub-problems where each multicast group can be optimized separately. In terms of LP, such a constraint is termed a complicating constraint. One of the ways to handle this kind of limitation is to apply some decomposition method.

Decomposition methods

Decompositions are computational methods that indirectly consider complicating constraints. Instead of solving the original compact problem including complicating constraints, two problems are solved iteratively: a simple so-called Master Problem (MP) and a problem similar to the original one but without complicating constraints. The MP is solved repetitively and complicating constraints are progressively taken into account [136]. This idea is formalized by the Dantzig-Wolfe decomposition [173].

As the author of [136] further explains, if complicating constraints are ignored, i.e., they are relaxed, the basic feasible solutions of the relaxed problem can be used to produce a feasible solution of the original (nonrelaxed) problem. A feasible solution is obtained

by solving the sub-problem that reflects the complicating constraints as dual variables[1] incorporated into the adjusted objective function, i.e., reduced cost. In case the minimized reduce cost is negative, a tentative feasible solution becomes a feasible solution to the relaxed problem. Any solution of the problem is a convex combination of the basic feasible solutions of the relaxed problem; therefore, it is itself a basic feasible solution of the relaxed problem. The MP is a restricted version of the original problem and, therefore, its objective function value is an upper bound of the optimal objective function value of the original problem.

Although the decomposition described above improves scalability, the cardinality of the set of feasible solutions becomes prohibitive for large instances, and it is demanding to state all the variables of the MP explicitly. At this point, the concept of Column Generation (CG) emerges. As noted in [173] the appealing idea of CG is to work only with a sufficiently meaningful subset of variables, forming the so-called Restricted Master Problem (RMP). More variables are added only when needed. As in the simplex method, a promising variable has to be found in every iteration to enter the basis.

In CG, each iteration consists of two following steps.

1. Optimization of the LP-relaxed RMP, in order to determine current optimal objective value and values of dual variables.

2. Optimizing a sub-problem, also known as a Pricing Problem (PP), to find out if there is still a feasible improving solution, i.e., the minimum reduced cost of PP is negative.

The main benefit is in work with a reasonably small subset of columns. The RMP optimally solves the MP as well. When dealing with a finite set, the column generation algorithm is exact [173].

Decomposed model

The idea behind the proposed model follows the concept of CG as described in the previous section. The complicating constraint (6.15) from the compact model was moved to MP constraints, while the rest of the compact model remains and becomes PP.

The multi-tree BDLDV CG model is built on the selection of a multicast tree, i.e., a Steiner tree, in the form of a forwarding configuration that is dynamically generated by PP. An instance of configuration c is selected from the set C of generated configurations.

[1]Each LP problem (primal) is associated with another LP problem called dual, and vice versa. This feature is described by Duality Theory, and it is often used to solve LP problems. It turns out that every feasible solution for one of these two linear programs gives a bound on the optimal objective function value for the other [140]. There is exactly one dual variable for each primal constraint and one dual constraint for each primal variable [138].

The MP objective function in (6.22) minimizes delay variation δ_T over all multicast groups $M \in \Gamma$, and is subject to constraints in (6.23)–(6.25).

$$\min \left\{ \sum_{c \in C} (\delta^c_{max} - \delta^c_{min})\, z_c \right\} \tag{6.22}$$

The fact that only one configuration is considered per multicast group is guaranteed by constraint (6.24), where $z_c \in \{0,1\}$ is a decision variable specifying whether configuration c is elected. The variable z_c plays identical role in the bandwidth constraint (6.23), where together with variable $y_{\ell,c} \in \{0,1\}$ reflect whether traffic is forwarded on link ℓ for particular configuration c. The auxiliary function $\mathbb{MCAST}$ points only to configurations $c \in C$ that are associated with a particular multicast group $M \in \Gamma$.

$$\sum_{M \in \Gamma} \sum_{c \in C | \mathbb{MCAST}(c) = M} z_c y_{\ell,c} \beta^M \leq c_\ell \quad \ell \in L \tag{6.23}$$

$$\sum_{c \in C | \mathbb{MCAST}(c) = M} z_c = 1 \qquad M \in \Gamma \tag{6.24}$$

$$z_c = \{0,1\} \qquad c \in C \tag{6.25}$$

The original objective function has to be adjusted for PP to complete the definition of the decomposed model. Loosely speaking, the PP objective function is an interface between RMP and the sub-problem dynamically generating potentially improving solutions. As noted in the previous section, this role is dedicated to dual variables in the definition of the reduced cost to minimize. In our case, the PP objective function in (6.26) incorporates dual variables $u^{(6.23)}$ and $u^{(6.24)}$ associated with constraints (6.23), and (6.24).

$$\min \left\{ (\delta_{max} - \delta_{min}) - u^{(6.24)} - \sum_{\ell \in L} u^{(6.23)} y_\ell \right\} \tag{6.26}$$

The PP problem is subject to constraints already defined for the MSPT and the compact multi-tree BDLDV problem. Namely, the flow conservation constraint in (6.7), tree aggregation constraints in (6.8)–(6.11), delay constraints in (6.13)–(6.14), and loop-prevention constraints (6.16)–(6.19).

6.1.5 Decomposed model workflow

Although the decomposition framework was outlined above, the execution plan of such a decomposed model is, in practice, always problem-specific. In the case of the multi-tree BDLDV problem, the proposed solution is derived from a solution for the SLE-RWA

Model published by M. Kozak in [174]. The original schema depicted in Figure 6.2 details the workflow of the model execution.

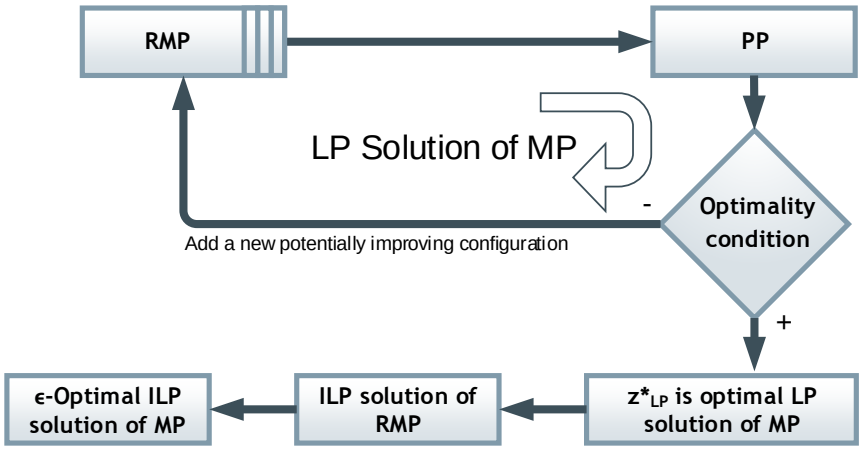


Figure 6.1: Original workflow *wf1* applied to the decomposed multi-tree BDLDV multicast model.

The process is usually initialized with a dummy solution. The main cycle follows the algorithm of the CG method. The MP is relaxed, then, PP is solved sequentially for each multicast group, and potentially improving configurations are added to set C of RMP. The relaxation of MP underlies the conversion from $z_c = \{0, 1\}$ to $0 \leq z_c \leq 1$. In case the optimality condition is fulfilled, i.e., none of the PP executions for each multicast group return a non-negative reduced cost, the LP optimum of $z^{\star}_{LP}$ is found. In the sequel, the *branch-and-bound* algorithm, see Section 4.1.1, is used to obtain the ILP solution of RMP. As the author of [174] notes, in practice, an integer solution that can be very close to the optimal integer value, or at least an integer value $\widetilde{z}_{ILP}$ for which accuracy can be estimated as $\epsilon = \frac{\widetilde{z}_{ILP} - z^{\star}_{LP}}{z^{\star}_{LP}}$.

Although original workflow *wf1* in Figure 6.1 is satisfactory, it does not incorporate the already optimized solutions based on the GA. Moreover, the optimality condition may cause the loss of sub-optimal, but feasible, solutions for large instances. In practice, the main cycle was unable to reach the optimum in the given time range available for optimization tasks. Consequently, when running the first workflow, all configurations were lost. To tackle this issue, the original workflow was enhanced as depicted in Figure 6.2. Using the second version of the workflow denoted *wf2*, the ILP solution is obtained in each iteration of the main cycle. The reason is straightforward. As the pre-generated configurations are of high quality already, it is beneficial to have the latest sub-optimal valid configurations at one's disposal rather than none of the optimal configurations.

An initial part of the second workflow is the pre-generation of initial configurations by

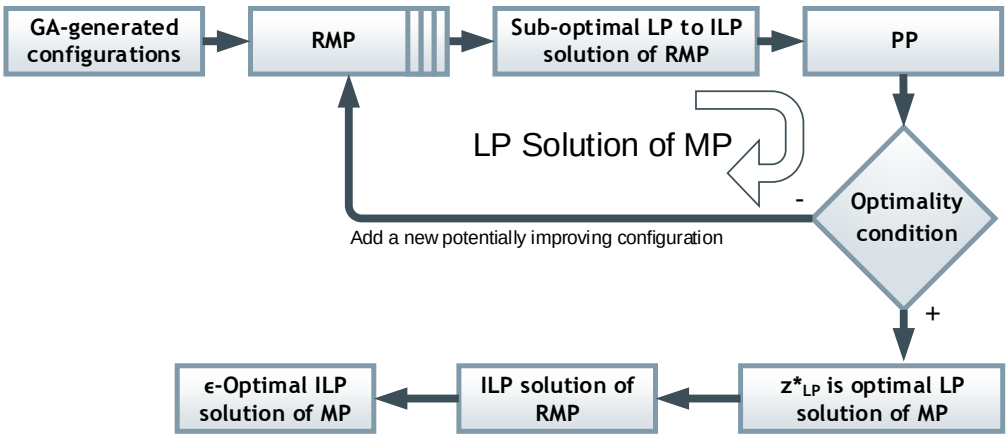


Figure 6.2: Enhanced workflow *wf2* applied to the decomposed multi-tree BDLDV multicast model.

the GA extensively presented in the previous chapter. These configurations are generated only once before the main cycle of the ILP program is executed. At least one configuration per multicast group and instance is always generated, but in practice other non-best top solutions are initially accepted to support the diversity of configurations. The GA is run repetitively to support the diversity and probability of the optimum LDV is being found. The number of repetitions is directly proportional to the number of links in the given problem instance. This approach allows the main cycle to start with nearly optimum configurations in the RMP stack of feasible solutions. Since sub-optimal non-best configurations are included, the diversity of C may help to overcome potential congestion problems caused by capacity limits at the very beginning of the model execution.

6.2 Evaluation

The evaluation part of this chapter is structured as follows. The first subsection focuses on the performance of the proposed multi-tree BDLDV models and their comparisons, to better understand the execution workflow shown in Figure 6.2, and to verify our design goals. In the sequel, the second subsection closes this chapter with detailed simulations, and eventually, the contribution of the proposed models is presented in the simulation results.

6.2.1 Model comparison

The evaluation of the proposed models is identical to the procedure presented in Section 4.2.2, though extended to fit the multi-tree problem. Since the ILP model for LDV

performed poorly, only 1000 instances were generated in total and under different conditions. At first, the number of nodes in graph instances was lowered to a range from 8 up to 24, the number of steps of the multicast coverage was decreased to 4 in a range from 0.2 to 0.8, but each instance contains multiple graphs ranging from 2 to 10 in the increments of 2. Preserving similar structural features as listed in Table 4.1, four different random models (Barabási-Albert, Dorogovtsev-Mendes, Erdős-Rènyi, Watts-Strogatz) were chosen concerning a sufficient variability of evaluated instances. Each of the graph instances was generated twice with the same parameters to support the diversity of the test set further.

The reference for the performance evaluation is the compact model detailed in Section 6.1.3, since the ILP model poses the performance baseline for potential enhancements by decomposition techniques. The compact model is the reference for the quality of results as well. All 491 of the 1000 instances successfully computed solutions using the compact model do not differ from the results reached by other decomposed models. The relative quality $R = 1$, and thus $\epsilon = 0$, implicates that solutions found by decomposed models are optimal. An example solution produced by the compact model is depicted in Figure 6.3.

Various criteria can be used to evaluate performance. The preferred one is the traditionally used speed up; however, the computational space available for optimization tasks was limited, and thus, each task was bounded by a time limit. The limit was 48 hours per task, where each task was run with the same configuration in the computational environment under approximately the same hardware conditions in NGI operated by MetaCentrum. Since the computation resources were similar, a measure of performance is considered to be

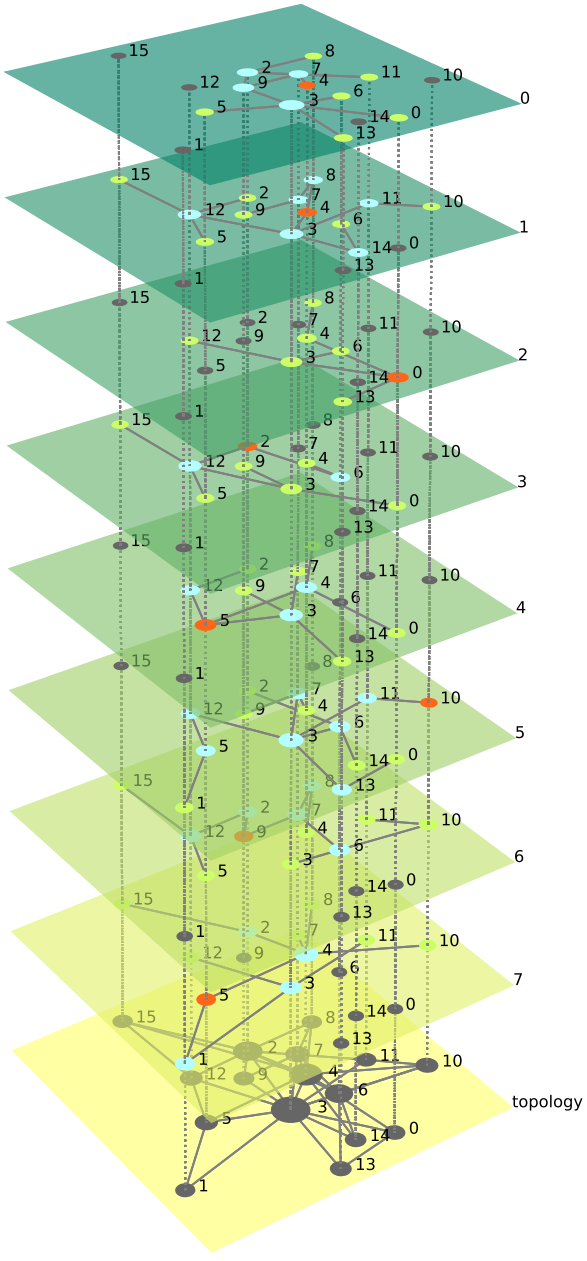
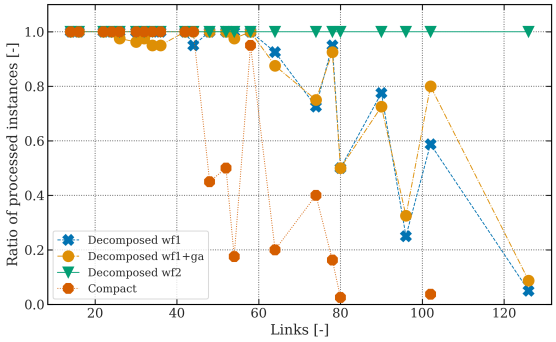


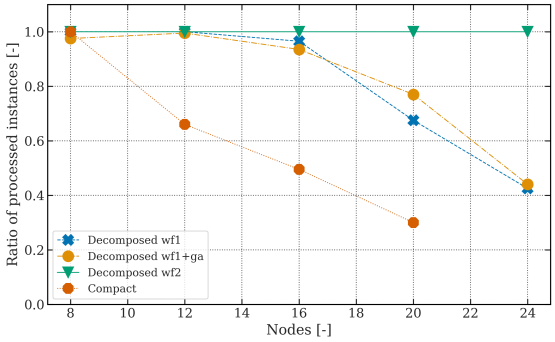
Figure 6.3: Optimal solution of a Barabási-Albert graph with 16 nodes, 0.4 multicast coverage, and 8 multicast groups.

the count of successfully computed optimization tasks using the presented models under the given time limit of 48 hours.

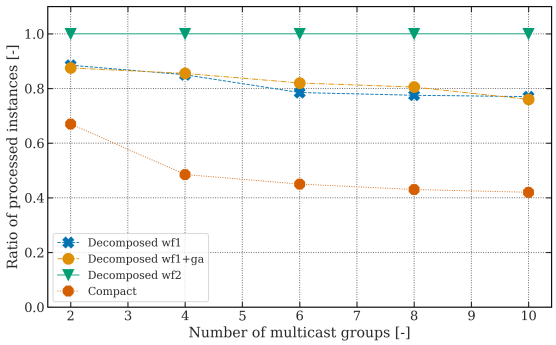
The performance evaluation using the above method is summarized in Figure 6.4. There are four models shown: the compact model, decomposed model *wf1* initialized by dummy solutions, decomposed model *wf1* initialized by GA-generated configurations, and the ultimate decomposed *wf2* model incorporating the GA-generated configurations as well. Each of the plots shows the expected observation that the compact model performs the worst. The last model *wf2* performs the best with full coverage of all instances, thanks to the ILP solution exported in every iteration of the main workflow cycle. As one can see for other models in both Figures 6.4b and 6.4a, the number of nodes, implicating the number of links, evinces the substantial dependency of computational complexity. While the compact model drops steeply right after the first group instances with only eight nodes, the decomposed models hold more relentlessly.



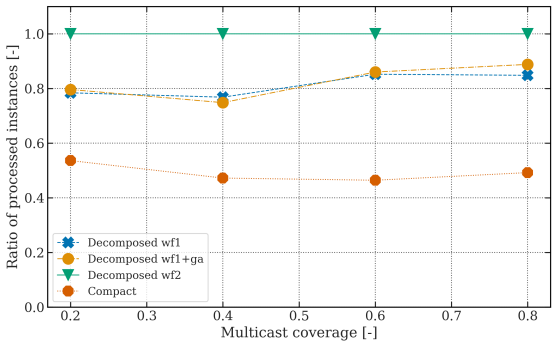
(a) The ratio of computed instances per group of instances containing a given number of links. The number of links is not spread equally due to the random graphs.



(b) The ratio of computed instances per group of instances containing a given number of links. The size of each group is equally distributed over 200 instances.



(c) The ratio of computed instances composed of a given number of multicast groups.



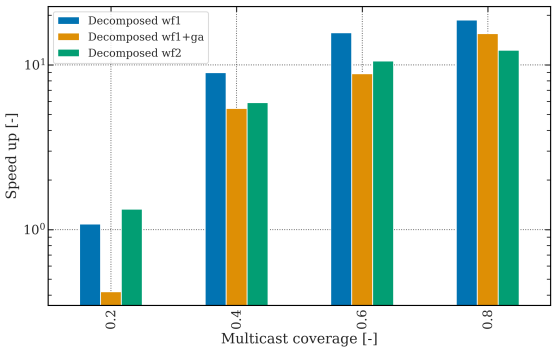
(d) Ratio of computed instances with given multicast coverage

Figure 6.4: The relative portion of successfully computed multi-tree BDLDV problem instances in the time window of 48 hours.

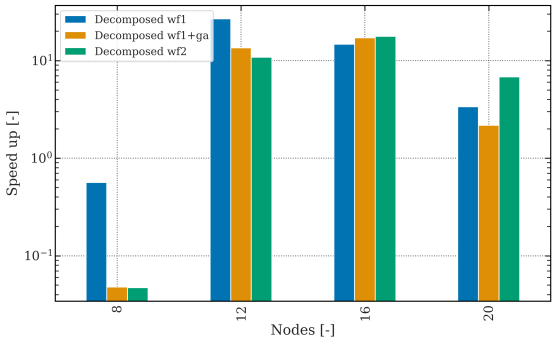
As depicted in Figure 6.4c, the number of multicast groups show a considerably smaller impact on the model performance in comparison to the number of nodes, or links. An

interesting observation emerges from Figure 6.4d, where the number of successfully computed instance slightly increases with the multicast coverage. Even though the decomposed model *wf1* initialized by the dummy or GA-based configuration does not show any significant difference in performance, the GA initialization is essential for the *wf2* model as at least one solution is composed of these initial configurations.

The second performance perspective depicted in Figure 6.5 is the speed up of the decomposed models in comparison to the baseline, i.e., the compact model. In contrast to the previous metrics, this comparison is limited only to those 491 instances successfully computed using the compact model. Despite this limitation, bar plots in Figures 6.5a and 6.5b clearly show that both small instances and sparsely covered networks are computed considerably slower using the decomposed models. This observation is not surprising as the overhead given by the complex workflow for decomposed models takes its toll. However, the very next step shows a significant difference in the speed up, where all three decomposed models perform almost similarly.



(a) Mean speed up of computed instances for various multicast coverages.



(b) Mean speed up of computed instances for instances containing a given number of nodes.

Figure 6.5: Speed up of all decomposed models in comparison to the compact model.

6.2.2 Evaluation workflow

Before we proceed to the simulations, it would be convenient, to sum up, what has been researched so far in this thesis. In the first and second chapters, the motivation behind the possible application of the multi-tree BDLDV multicast problem in the area of SG using a specific data stream of SVs, as defined by the IEC 61850 standard, was introduced. Results obtained from the numerical evaluation of the ILP model for the LDV problem showed its ability to truly lower the delay variation in the multicast tree in comparison to the trivial SPT approach. However, the scalability issue led us to the GA-based metaheuristic that under extensive evaluation shows the massive speed up of the task while preserving

the high quality of solutions. All these findings were put together to form the proposed decomposed model *wf2*, as presented and numerically evaluated above.

Although the numerical evaluation answers the question of optimality in the discrete domain, the nature of the real world is stochastic, and thus, the simulation of the optimized forwarding configurations is a natural format of the subsequent evaluation.

The evaluation workflow in Figure 6.6, partly inspired by the methodology published in [175], was split into three more or less separated groups of actions to achieve the simulation results we prefer. The first part, termed Inputs, covers the problem formulation that is later translated to a particular notation and, then, problem instances. These instances were generated following rules similar to the ones in Section 4.2.2. The main difference is that the random graphs were omitted in favor of common redundant LAN topologies, as introduced in Section 2.4.2, to adequately reflect real-world applications.

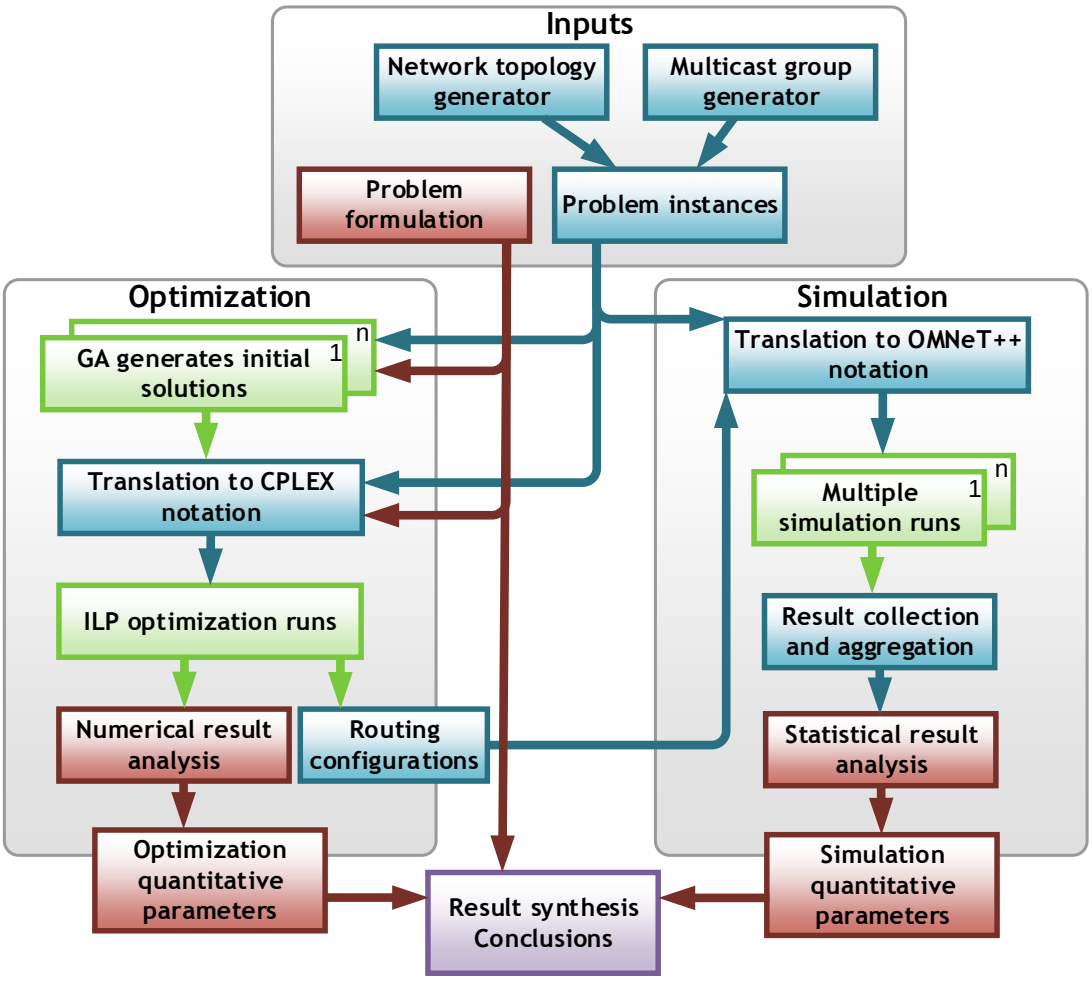


Figure 6.6: Complete evaluation workflow.

Each problem instance consisted of 10 to 30 multicast groups in increments of 5 groups and uniformly placed subscribers in a range from 20 % to 80 % of total access-level nodes. Each configuration was randomly generated five times to add additional diversity to the

evaluation set. The multicast publishers were randomly connected to the access-level nodes. In contrast to the random graphs, the network size was given by the kind of LAN topology and its regularity. Thus, the absolute number of nodes may differ among instances, but the relative multicast coverage is preserved.

To make the simulations as real as possible, the simulation model implements all latencies identified in Chapter 3 that affect the end-to-end delay: switch fabric latency, the wireline latency, and queuing latency. The switch node was modeled following the OQ architecture. The incoming frame is deserialized and directly placed to the appropriate output queue right after the switch fabric latency elapses. A wireline latency was randomly assigned to each link in a range from 50 to 500 ns. This latency is approximately proportional to the delay on Ethernet segments with lengths from 10 to 100 m. Switch fabric latency was randomly assigned to each node in a range from 800 ns to 1500 ns depending on the simulated data rate and reflecting the measurements in Section 3.2.4.

The generated instances were used in both the Optimization and Simulation part of the workflow in Figure 6.6. After the inputs were ready, the multi-tree BDLDV problem was optimized on these instances translated to CPLEX notation. Subsequently, the proposed GA was employed to find initial configurations passed to the decomposed model *wf2* as already mentioned above. This ILP model was optimized on given instances using the CPLEX Optimizer. Following the Optimization part of the workflow, these results were used for numerical evaluation of the model, but the forwarding results were transferred to the Simulation part.

Simulation process

The simulation depends on generated instances and forwarding information acquired from the results of the discrete optimization. Both inputs need to be translated into a specific OMNeT++[2] notation. While OMNeT++'s language called NED defines the instances, the forwarding information needed a custom XML-based description. Although the essential functionality is already part of the OMNeT++ framework and the INET framework adds the Ethernet switch basics [177], multicast components (source, sink and switch) had to be implemented from scratch or adapted accordingly.

Even though the SDN control seems to be overlooked in the description of the simu-

[2]OMNeT++ is a C++-based discrete event simulator for modeling communication networks, multiprocessors and other distributed or parallel systems [176]. OMNeT++ is open-source providing a very good API allowing scalable event-driven simulations to be created. The event-driven approach is natural to communication protocols as these are finite state machines that change their states in time. OMNeT++ represents a framework approach where the simulated domain is supported by a specific modular library that can be further adapted for the required simulation purpose. As fittingly noted in [174], the best approach for writing a module is following the UNIX philosophy which emphasizes building short, simple, clear, modular and extensible code that can be easily maintained and reused.

lation scenarios, SDN is the tool to install the forwarding configurations into FIBs along the multicast tree. Since these configurations are pre-computed in a proactive manner, there is no need to simulate the distribution from the SDN controller to SDN-enabled switches.

As instances were generated randomly, a situation when there were no feasible solution for the multi-tree BDLDV problem could occur with non-zero probability. This situation was common, especially for the optimization of instances with a high number of multicast groups sharing the 100Mbps infrastructure. For example, 30 multicast groups sharing one link requires $30 \times 4.416 \simeq 132.48$ Mbps, see Section 2.2.4 for details. For comparison purposes, a forwarding was computed for each instance using Kruskal's algorithm for MST, and then, optimized using the MSPT model and the decomposed multi-tree BDLDV model in the second workflow *wf2*. These results were initially obtained for 1 Gbps, but the very same results were used for 100 Mbps as well, to examine the impact of link saturation on the overall performance. As a complement to the previous optimization tasks, additional configurations were computed for 100 Mbps, but limited to 60% of the link capacity[3].

In the subsequent step, each network instance was repeatedly simulated under different conditions to reach a statistically significant amount of results for further analysis. The number of repetitions was set to 20. Another condition was an additional uniformly spread 0, 500, and 5000 ns to the generated multicast stream at each publisher, to mimic real-world inaccuracies and examine its impact on the BDLDV model contribution. Considering the simulation scenarios above, the total number of simulations resulted in 224 000 runs.

Simulation scenarios were run considering only the multicast traffic and omitting any background communication originating in other IEC 61850 applications. The reason is based on two assumptions, partly detailed in Section 2.2.4. At first, there does not exist any reliable pattern of such background communication, since it is heavily affected by a particular SAS deployment. Secondly, the generated high-priority multicast stream is tagged by an appropriate Priority Code Point (PCP) in the Ethernet header, and as such, it should be handled by the switch preferentially minimizing blocking at the egress port.

6.2.3 Simulation results

The analysis of the simulation results focuses on the following tasks in the time domain. Do the optimized configurations yield some improvement in comparison to the trivial MSPT or the conventional MST forwarding? What is the impact of optimized forwarding

[3]Our simulations on the switch queuing delay indicated that the limit 0.6 Erlang is a breaking point after which the delay steeply increases.

configurations on the crucial end-to-end delay limits, or does the jitter increase unacceptably?

Evaluation criterion

Since the objective of minimizing the delay variation δ_T, as defined in expression (6.3), produces absolute values unique for each problem instance, there was a need to define a metric that allows obtained results to be compared. A new improvement metric I_{ig} expressed in (6.27) was defined.

$$I_{ig} = \frac{med_{R_{MST}}(med_G(\{\delta_T^{ig,r}\}))}{med_{R_{ALG}}(med_G(\{\delta_T^{ig,r}\}))} \qquad ig \in G, r \in R \tag{6.27}$$

As we cope with multiple groups per instance and multiple messages per simulated multicast stream, median med_R of medians med_G of δ_T recorded in a set of all simulation runs R was chosen as a convenient approach to evaluate the optimization contribution. Improvement I_{ig} is then simply a ratio of δ_T medians per problem instance i and group g, from a set of ig pairs G, obtained for a base algorithm to values obtained from an optimization algorithm. The base algorithm is undoubtedly the MST as it must serve all nodes in the network, and the evaluated algorithms ALG are naturally MSPT and multi-tree BDLDV. The interpretation of the metric is straightforward and analogous to the speed up defined in (5.13). If the ratio is higher than 1 the ALG generated an improving configuration from the perspective of the objective function. The higher the value is the better the ALG performs.

Other metrics come directly from absolute values as it is not necessary to normalize them for comparison purposes.

Improvement

As mentioned above, two general scenarios are compared depending on the data rates of the simulated network infrastructure. Naturally, the 100Mbps link poses a potential bottleneck more likely than the 1Gbps network, especially for the higher number of multicast groups (25, 30). The difference is evident from Table 6.1. While the higher data rate hits almost 100% of successfully simulated instances, the latter 100Mbps data rate shows a significant drop in the ratio, especially as MST and the multi-tree BDLDV limited to 0.6 Erlang. While the MST drop is caused mainly by the excessive utilization of some links, implicating an infinite growth of frames queued on egress ports, the BDLDV 0.6E decrease is caused by infeasible solutions, i.e., there was no such forwarding configuration

fulfilling the given constraints. Simulations without forwarding configurations were not executed.

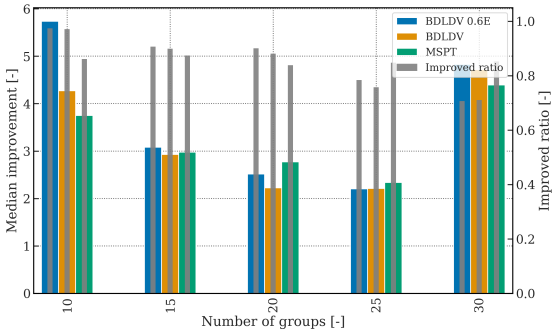
Table 6.1: The ratio of successfully computed simulations.

Mode	Algorithm	Simulation ratio [-]
1Gbps	MST	0.89
	MSPT	0.97
	BDLDV	0.97
100Mbps	MST	0.62
	MSPT	0.97
	BDLDV	0.88
	BDLDV 0.6E	0.65

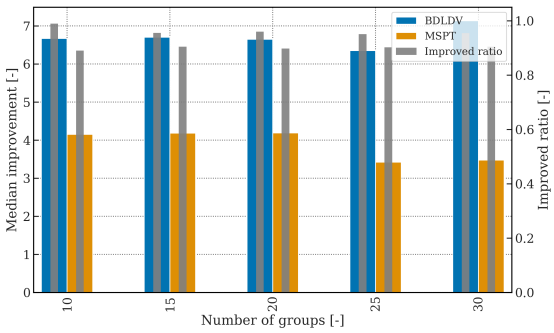
Considering improving solutions, i.e., simulations with $I_{ig} > 1$, we can inspect results under various cuts in Figure 6.7. The plots combine two metrics: the improvement and share of improving solutions. From the global perspective, multi-tree BDLDV solutions to 90 % and MST 85 % at 100 Mbps, and at 1 Gbps the multi-tree BDLDV hits 96 % and MSPT 90 %. These numbers clearly show that both trivial and complex algorithms outperform MST as both consider only a particular subset of nodes in the network.

Although the overall numbers seem encouraging, the scale of improvement differs greatly depending on instance features. Surprisingly, the limited BDLDV to 0.6 Erlang does not show better results than the BDLDV configurations with no limits in most cases at 100 Mbps as one can see in Figures 6.7a and 6.7c. However, this is most likely because the plots show results filtered for common instance basis, i.e., such instances that were successfully simulated with all algorithms. The limited and non-limited versions probably constructed similar forwarding configurations in most cases as the improvement is only slightly better for the limited BDLDV. As the plot in Figure 6.7a shows, the improvement is decreasing with the number of multicast groups with an increase at 30 groups, possibly caused by the increasing saturation of MST configurations. This trend is almost negligible in Figure 6.7b, but one can notice the constant share of improved solutions. This claim remains, even in comparison with different multicast group sizes in Figure 6.7d.

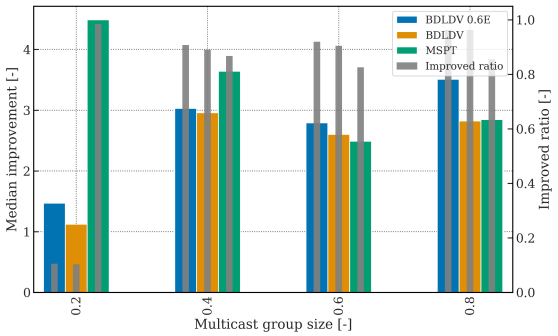
While the results at 100 Mbps show only a few exceptions where BDLDV and MSPT results differ significantly, the unsaturated 1 Gbps show that the problem-aware multi-tree BDLDV outperforms MSPT at all cases. The difference is apparent in Figure 6.7f at topologies *ft*, *ttg*, and *ctg9*. Even though these topologies are highly-redundant, it is beneficial for our optimization problem. However, the *ctg10* is highly-redundant as well but does not outperform MST considerably. The difference may lie in the network structure at the second level, where worse performing topologies are connected in a full ring while the better topologies are not. A similar observation for BDLDV is valid at 100Mbps as well, but MSPT plays its part in some cases.



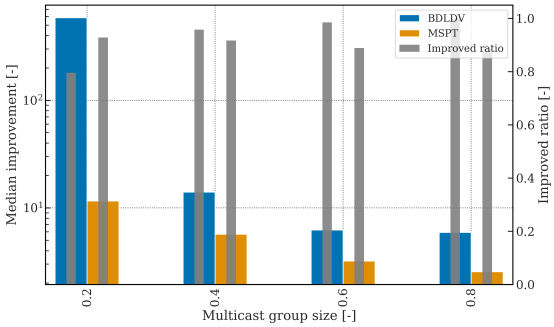
(a) Median improvement on a 100Mbps network for a given number of multicast groups.



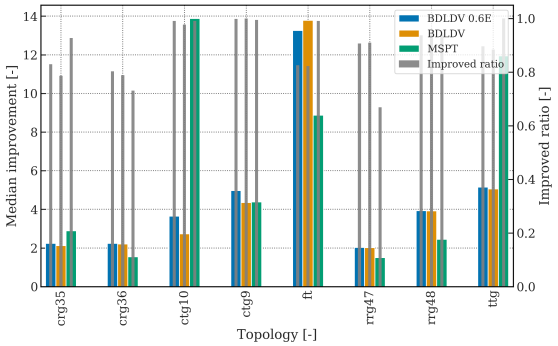
(b) Median improvement on a 1Gbps network for a given number of multicast groups.



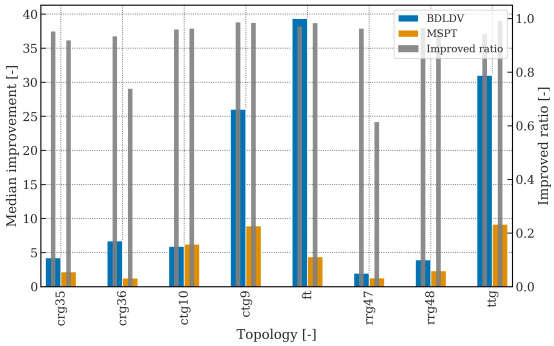
(c) Median improvement on a 100Mbps network for a given multicast group size.



(d) Median improvement on a 1Gbps network for a given multicast group size.



(e) Median improvement on a 100Mbps network for a given topology.



(f) Median improvement on a 1Gbps network for a given topology.

Figure 6.7: Comparison of improvements and share of improving solutions.

End-to-end delay

From the perspective of the objective function, simulations show that even under the limited diversity of the underlying infrastructure the multi-tree BDLDV model exceeds a boundary of 90 percentage points of improving solutions in comparison to the traditional MST approach. Even though these results are encouraging, the crucial constraint of maximum transfer time equal to 3 ms, as defined in Section 2.2.3, cannot be omitted.

The definition of the maximum transfer time implies the implementation of the end-to-end delay measurement in the simulation model, where time is measured from the point where a generated multicast message leaves the application layer until the point the

Table 6.2: End-to-end delay recorded for all simulated algorithms and data rates.

Mode	Algorithm	Max e2ed [ms]	Median e2ed [ms]
1Gbps	MST	0.033	0.011
	MSPT	0.032	0.007
	BDLDV	0.036	0.011
100Mbps	MST	177.446	0.077
	MSPT	109.948	0.051
	BDLDV	160.429	0.072
	BDLDV 0.6E	0.299	0.069

Table 6.3: Maximum end-to-end delay recorded for 100Mbps.

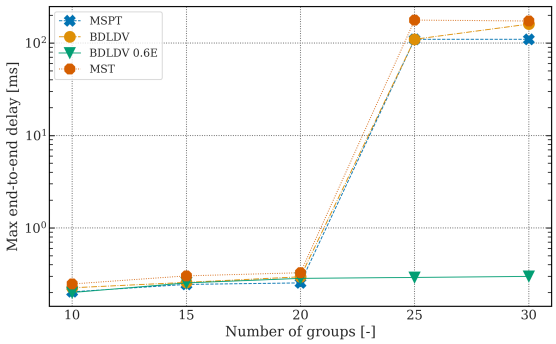
Groups	Algorithm BDLDV 0.6E | BDLDV | MSPT | MST Max end-to-end delay [ms]			
10	0.200	0.226	0.204	0.248
15	0.255	0.258	0.245	0.303
20	0.285	0.297	0.255	0.328
25	0.292	109.988	109.933	177.446
30	0.299	160.429	109.948	173.216

message duplicate enters the application layer at a particular multicast subscriber. Thus, the terms transfer time and end-to-end delay are interchangeable in the following text.

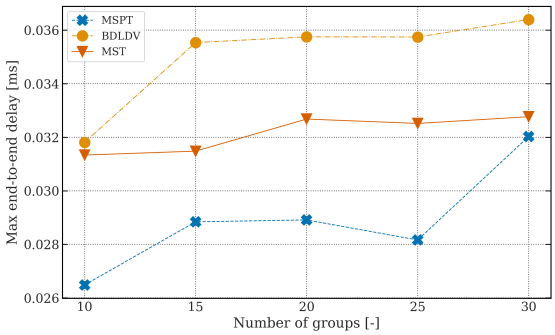
The overall results are listed in Table 6.2. From the standardization perspective, the critical measure is the maximum value which is the highest recorded end-to-end delay during all simulation runs. While results for 1 Gbps clearly show that the capacity of links was not saturated with maximum values slightly surpassing 0.03 ms, maximum times reported for 100 Mbps exceeds the limit significantly. The only exception is the BDLDV model limited to 0.6 Erlang where the infeasible instances were skipped.

The saturation issue at 100 Mbps is detailed in Table 6.3. The critical breaking point is at 25 multicast groups when the total bandwidth produced by publishers breaks the capacity with $\simeq$ 110.4 Mbps. It is not surprising that the unlimited BDLDV model reports unsatisfactory values as well since it was initially optimized for the 1Gbps data rate and simulated at 100 Mbps for comparison purposes. Applying the proper bandwidth limit confirms the anticipated results delivered by the proposed decomposed multi-tree BDLDV model. Using the adequately parameterized BDLDV model, instances with excessive capacity demands can be easily identified.

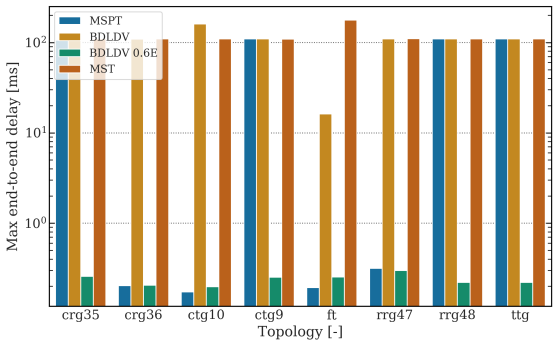
Generally, the maximum transfer time slightly grows with the number of groups at all considered algorithms as depicted in Figures 6.8a and 6.8b. The plot for the 100Mbps data rate shows once again the steep increase after 20 groups when the link capacity is saturated and delay on egress ports along the way from the multicast publisher to subscribers grows above all limits. As the moderate increase of the maximum transfer time with the number of groups is observable at all algorithms, it is likely caused by the



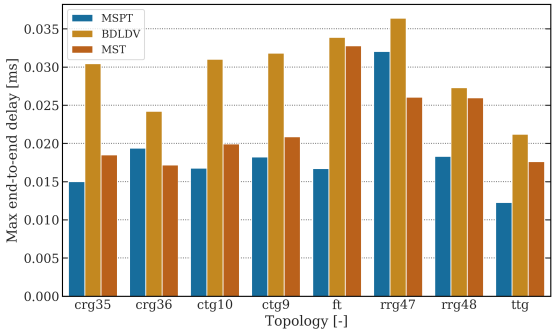
(a) Maximum end-to-end delay on a 100Mbps network for a given number of multicast groups.



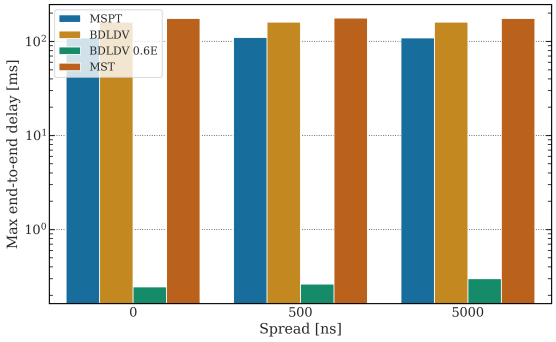
(b) Maximum end-to-end delay on a 1Gbps network for a given number of multicast groups.



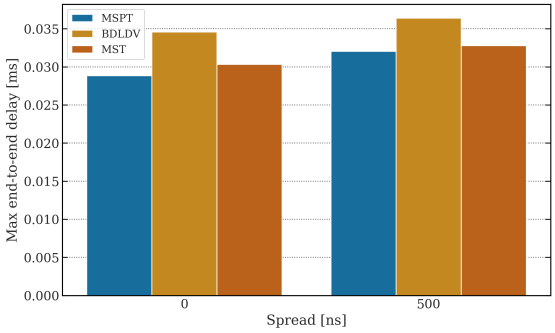
(c) Maximum end-to-end delay on a 100Mbps network for a given topology.



(d) Maximum end-to-end delay on a 1Gbps network for a given topology.



(e) Maximum end-to-end delay on a 100Mbps network for a given time spread during message generation.



(f) Maximum end-to-end delay on a 1Gbps network for a given time spread during message generation.

Figure 6.8: Comparison of maximum recorded transfer times of generated multicast messages under different conditions.

increasing service time at egress ports. Although each topology instance was generated with various link lengths and switch fabric latencies, guaranteeing a certain degree of randomness of message distribution in time at switch nodes, we decided to examine the impact of non-coherent multicast streams. It turns out that timely spread generation of multicast streams show a negligible impact on the maximum transfer time as depicted in Figures 6.8e or 6.8f.

Analogous to the improvement metric presented above, there is an evident relation

between maximum transfer time and the kind of topology of the particular instance. Results for 1 Gbps, depicted in Figure 6.8d, demonstrate higher values by the BDLDV model as it constructs larger trees to fulfill the objective. On the other hand results for 100 Mbps depicted in Figure 6.8c are negatively affected by the link saturation with the exception of the limited BDLDV model and the MST at half of the simulated topologies.

Jitter

The last of the metrics from the time domain is jitter, which can be potentially negatively influenced by larger trees produced by the LDV objective. Although there is not any standardized requirement in the area of jitter, it may affect some applications, and generally, the lower the jitter, the better it is for streamed services. In this thesis, the term packet delay variation is not used[4], since such a notion may be confusing in the context of the LDV problem. Hence, the jitter obtained from simulations was computed as the difference between the maximum and minimum end-to-end delay on a particular subscriber over the whole simulated period. This method expresses the worst case scenario.

Since the goal is to analyze the impact of proposed models on jitter, only simulations with a coherent multicast stream were considered. Otherwise, the non-zero time spread at message generation would negatively affect presented results. It turned out that the only a small fraction of the simulated instances recorded non-zero jitter values as listed in Table 6.4.

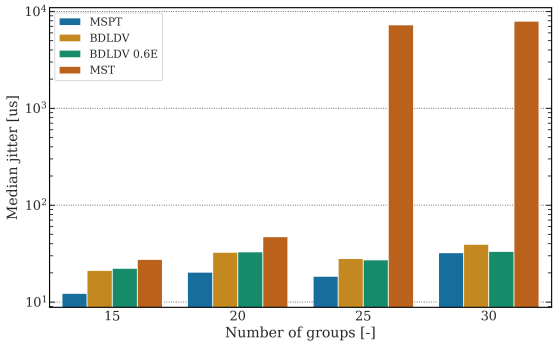
Table 6.4: The ratio of simulated instances with non-zero jitter.

Mode	Algorithm	Ratio [-]
1Gbps	MST	0.059
	MSPT	0.058
	BDLDV	0.056
100Mbps	MST	0.116
	MSPT	0.062
	BDLDV	0.082
	BDLDV 0.6E	0.058

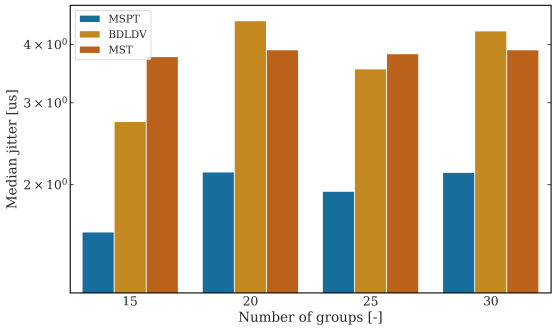
That the limited amount of non-zero jitter instances and the share of such instances is relatively similar indicates that only a few and common instances negatively affect jitter values. Analogous to the improvement metric defined in (6.27), values plotted in Figure 6.9 are the median of medians over multicast groups and simulation runs.

As one can see in Figures 6.9e, 6.9a and 6.9c for 100 Mbps data rate only the MST significantly surpasses the others. This phenomenon is likely caused by saturated links at instances based on topologies *crg36*, *ctg10*, *rrg47* and *rrg48* with 25 and 30 groups and

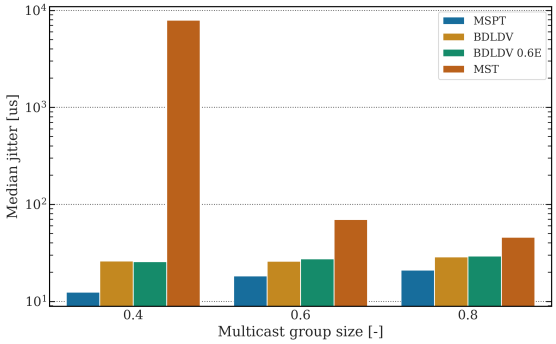
[4]The packet delay variation is often defined as the difference in end-to-end delay between selected packets at a particular host.



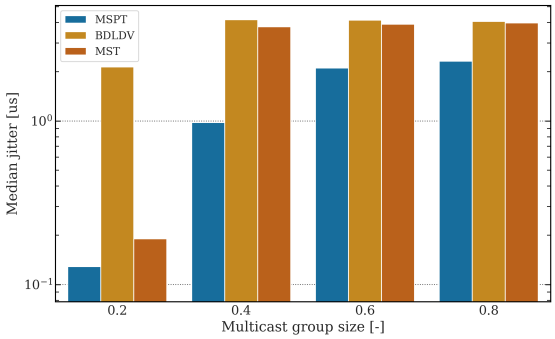
(a) Median jitter on a 100Mbps network for a given number of multicast groups.



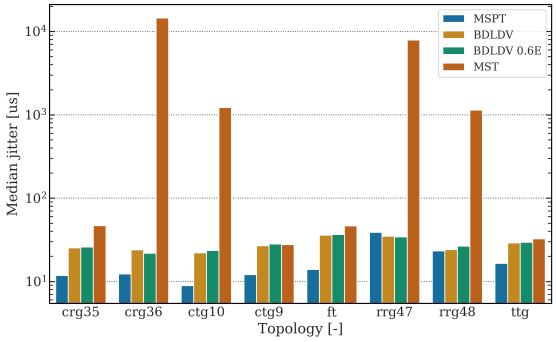
(b) Median jitter on a 1Gbps network for a given number of multicast groups.



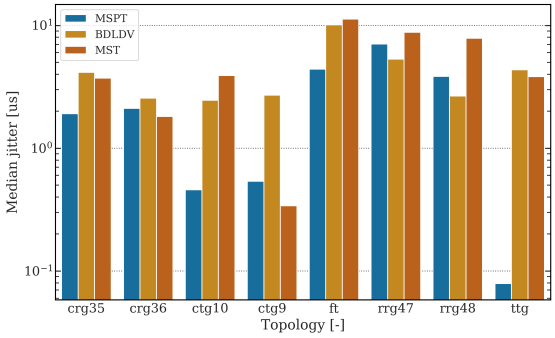
(c) Median jitter on a 100Mbps network for a given time spread during message generation.



(d) Median jitter on a 1Gbps network for a given time spread during message generation.



(e) Median jitter on a 100Mbps network for a given topology.



(f) Median jitter on a 1Gbps network for a given topology.

Figure 6.9: Comparison of non-zero jitter of messages delivered to particular subscribers and generated with a zero time spread under different conditions.

under 0.4 multicast coverage. The MST likely produced a forwarding tree resulting in a serious traffic bottleneck. Although the results presented for the end-to-end delay contain evident bottlenecks as well, the median is more favorable to jitter results directly pointing to the truly problematic instances hitting millisecond values.

The 1 Gbps in Figures 6.9f, 6.9b and 6.9d demonstrate more conservative numbers in order of microseconds. Considering the amount of non-zero jitter instances, these numbers are fairly low. The most important observation is that even though the proposed multi-

tree BDLDV model tends to produce larger forwarding trees, simulation results disprove the concerns with extensively increasing jitter.

6.2.4 Evaluation summary

The last chapter offers a solution to the limited scalability of the original ILP model for the LDV problem described in Section 4.2. Although the presented multi-tree multicast problem makes the task even more complicated, the decomposed BDLDV based on RMP and CG employing GA-based initial solutions is able to deliver high-quality solutions for real-world-sized network instances. Focusing on the optimality criterion, the decomposed BDLDV produced solutions of the same quality as the compact ILP model. Even though the optimality is not guaranteed at larger instances by any means, the stability of the proposed model is excellent.

The next step of evaluation, based on simulations, show several observations, but the most important one is that the objective of the minimization of the delay variation is possible even on the non-deterministic Ethernet. The share of improved instances, in comparison to traditional MST algorithm, exceeded 90 % in the vast majority of simulations on common redundant network topologies. As results for the 100Mbps infrastructure show, the proposed multi-tree BDLDV model cuts off infeasible problem instances whose communication requirements oversubscribe available network resources. In the case of the 1Gbps infrastructure, the link capacity created sufficient space to unleash the potential of our optimization fully, and it outperformed the trivial MSPT approach. The adequately dimensioned network capacity allowed the strict IEC 61850 requirement on maximum transfer time of SV messages to be fulfilled. The unsaturated problem instances complied with the 3 ms limit without problems as it reached only 0.036 ms in the worst recorded case. Simultaneously, the concerns about substantial jitter values were disproved as only 5 % to 10 % of instances reported non-zero jitter, which, in the case of the BDLDV solutions, was not excessive in comparison to other approaches.

Just because something doesn't do what you planned it to do doesn't mean it's useless.

Thomas Alva Edison

7 Conclusion

As the doctoral thesis straddles several topics, the following sections sum up how are the problems related and what results have been achieved. Then, a fulfillment of thesis aims is appraised, and potential future research is briefly outlined.

7.1 Thesis summary

The core idea behind this thesis arose from several years of working in different technological areas that are non-related t first glance: SG communication protocols, the SDN concept and computer science. As the promising IEC 61850 standard was getting attention from power utility and distribution companies, it turned out that it posed a research opportunity. Although the second edition corrected and simplified the standard deployment, some optimization tasks remained open. One of such task was the optimization of L2 multicast forwarding in local SA networks, especially the continuous stream of data produced by MU, known as SV consumed by various IEDs in the network. Each such stream of SVs can consume up to 10 Mbps bandwidth and in the case of the commonly used MST forming the forwarding topology a higher number of multicast groups could saturate the capacity of active links, though the capacity of the redundant network connections remained underutilized. However, to utilize all links in the network for a custom-tailored forwarding was in contradiction with MST-based protocols. The answer was the SDN

concept that detaches the control and forwarding plane of networking devices shifting network intelligence towards the logically centralized controller.

Having the problem application area and a tool to propagate the optimized forwarding, the problem abstraction and formalization was only a step ahead. The minimum multicast tree problem is generally known as the minimum Steiner tree in networks, and it is from the $\mathcal{NPO}$ class. This essential problem is often modified making it even harder to solve, as detailed in Section 2.5.2, and in the case of this thesis, the focus was put on the specific type of the multicast problem named Delay and Delay Variation Multicast problem that is $\mathcal{NP}$-complete. First, the research of the optimization task started with a less complex LDV problem using ILP to solve the problem on random network topologies. The numerical results confirmed that the proposed model minimized delay variation of the multicast tree in comparison to the naive SPT approach. From the structural perspective, results further showed that the optimized trees tended to be composed of a higher number of links. The evaluation of the proposed ILP demonstrated a scalability barrier. Since the problem is classified as $\mathcal{NPO}$, the observation was not surprising and supported the decision to step away from the exact ILP solutions towards the heuristic approach.

Results achieved by metaheuristics on real-life problems offered proof of the ability to succeed where other more rigid methods failed, and thus, further research focused on evolutionary algorithms, particularly on GA. The extensive evaluation of different genetic operators, involving novel crossover and mutation operators, have shown that the proposed algorithm reaches best results under high mutation rates. Due to the nature of the LDV problem where the optimal solution may be completely isolated and discontinued from other solutions, the intensive mutation empowers the exploration phase of the algorithm keeping the population diverse enough. In comparison to the ILP model for the LDV problem, the proposed GA performs exceptionally well with a reported speed up of lower hundreds while keeping a high ratio of optimal solutions reaching 90 %. The Bayesian optimization was applied to find the best hyperparameters in companion with scenario-based optimization, resulting in similar results.

The encouraging results delivered by the proposed GA were taken into account in the last part of this thesis connecting all chapters. The ultimate BDLDV problem was defined by adding the delay and bandwidth constraints to the multicast multi-tree problem where multiple multicast groups share a common network topology. The lesson regarding the scalability learned on the ILP model motivated the author of this thesis to employ a decomposition technique built on the RMP approach in combination with CG and based on initial solutions from GA. After the proper model workflow was identified, the numerical evaluation showed that the decomposed model is superior to the compact ILP model. The decomposed model is stable in quality of solutions while reaching upwards of

tenfold speed at large instances.

However, the numerical evaluation is only one part; the other is the simulation model that mirrors the real-world nature of the non-deterministic Ethernet, the communication channel for the L2 multicast. The simulations were run on redundant LAN topologies to be close to real use-cases in substations. Simulation models were generated with switch fabric latencies obtained for SDN-enabled switches by the measurement methodology presented in Chapter 3 to make the simulation even more realistic in the time domain. The simulation results prove that the BDLDV optimized forwarding fulfills its objective under specific conditions. In situations where there was enough bandwidth available, like the 1Gbps infrastructure, the delay variation was truly minimized in comparison to the naive MSPT approach, and over 90 % of instances was improved to traditional MST. When the network capacity was not sufficient, the proposed model directly marked such instances as infeasible. The overall simulation results show that when the network is free of saturated links, the crucial requirement of a maximum transfer time set to 3 ms is met and jitter does not grow noticeably using the BDLDV optimized forwarding.

7.2 Fulfillment of the thesis aims

Aim 1 Describe the potential application area of the optimization algorithm in the context of contemporary Smart Grid standards.

The analysis of the IEC 61850 standard in Chapter 2 revealed a deficiency of the multicast forwarding in substation networks using the Ethernet. Conventional techniques rely on distributed algorithms potentialy underutilizing redundant links. The SDN approach to control data flows in the network allows the realization of more advanced forwarding logic in the network. Using the SDN concept, it is possible to implement an optimized multicast tree, a Steiner tree in networks, with additional constraints. The first step is the LDV problem, defined in 4.2, being a prerequisite to the more complex multi-tree BDLDV problem, defined in Section 6.1.1.

Aim 2 Design a methodology to measure switching latency of Ethernet switches operating at high data rates.

The first contribution to the evaluation of the ultimate multi-tree BDLDV algorithm is a methodology detailed in Chapter 3. The methodology was subsequently published in peer-reviewed journals. The proposed methodology allowed us to measure reliably and semi-automatically switch fabric latency even at 10Gbps switches, hitting hundreds of nanoseconds at the SDN-enabled switch. Although latency was measured only for Ethernet switches, the methodology is not limited to this tech-

nology, and it can be adapted for other technologies by using proper interfaces.

Aim 3 Formulate a LDV optimization problem for a single multicast group, and propose an optimization algorithm delivering exact solutions.

The LDV problem is formulated in Section 4.2, together with the underlying network model. Since the problem contains only linear constraints, the problem was formulated as an ILP program and solved by the CPLEX Optimizer to obtain the exact solutions. Even though the obtained results were optimal, the maximum size of problem instances was limited to 20 nodes. Larger instances exceeded the time limit of 48 hours available for computing in the Metacentrum cluster. Nonetheless, results obtained from the numerical evaluation show that the ILP model for LDV optimizes the delay variation in comparison to the naive approach of the SPT model. Generally, the delay variation is lower at smaller multicast groups, while the end-to-end delay and tree size are larger. As the multicast coverage grows, results for LDV and SPT models approach each other, since the number of link combinations are becoming smaller.

Aim 4 Design a metaheuristic for the single multicast group LDV problem and compare results with the exact algorithm proposed in Aim 3.

Scalability is the main drawback of the proposed ILP model for the LDV problem. The metaheuristic approach was chosen to tackle this issue, particularly the GA detailed in Chapter 5. This evolutionary algorithm is well researched and while the reasoning behind each phase of the algorithm is not so straightforward as in the case of the simplex method for ILP programs, it can deliver high-quality results. Although it takes much effort to tune the whole algorithm regarding graph-specific operators and hyperparameters, the speed up to the exact ILP model was frequently a hundredfold, especially at large instances, while the quality of results was comparable to exact solutions. The results from the hyperparameter optimization show that the GA can find an optimal solution for nearly 90 % of the evaluated problem instances under proper setup.

Aim 5 Formulate a constrained BDLDV problem for a multiple group multicast sharing common topology. Propose and evaluate an algorithm that delivers nearly-optimal solutions for real-world-sized instances.

As the LDV problem was only a starting point, the more complex problem reflecting the true nature of SA networks with multiple multicast groups emerged. The multi-tree BDLDV problem formulated in Section 6.1.1 comes from the LDV problem but adds the delay and bandwidth constraints. Analogous to the LDV problem, the

first step was to define a reference ILP model, i.e., a compact model. Although the model delivers optimal solutions, scalability was, unsurprisingly, even worse than in the case of the LDV problem.

A decomposition technique had to be employed to tackle the scalability issue of the multi-tree BDLDV problem. The model was split into RMP taking care of the objective function and the bandwidth constraint, and PP supplying the RMP with forwarding configurations valid in the time domain and minimized via dual variables. Although the decomposed model produced optimal solutions in comparison to results obtained from the compact model, the number of computed instances was not significantly better. After adding initial configurations generated by the GA and relaxing the problem in each RMP iteration, the scalability changed dramatically. All problem instances based on random graphs, detailed in Section 6.2.1, were optimized and obtained results were optimal in comparison to results from the compact model.

Aim 6 Implement simulations on redundant topologies using the measured switch fabric latencies and apply the optimized multicast forwarding configurations. Verify the results and compare all algorithms under different conditions.

The simulation model described in Section 6.2.2 was built in the event-driven simulator OMNeT++. The model conscientiously reflects all components participating in the end-to-end delay as defined in Section 3.1: wireline latency, store-and-forward latency, switch fabric latency, and queuing latency. New problem instances on redundant topologies conceivable in SA networks, presented in Section 2.4.1, were generated using the measured switch fabric latencies from Section 3.2.4. The stochastic simulation was run repeatedly on all 800 instances for each of the compared algorithms to reach statistically significant data.

Results show that in terms of the objective function, the decomposed model for multi-tree BDLDV produces optimized forwarding configurations that, in most cases, outperforms both the naive MSPT approach and the conventional MST.The improvement level is dependent on the available bandwidth in the network, where the higher data rates perform naturally better than lower ones. At problem instances, for which the multi-tree BDLDV model produced an optimized solution, i.e., there were no saturated links, the reported end-to-end delay was far below the required 3 ms and jitter was reported only at 5 % to 10 % of received packets with an insignificant difference to other algorithms.

7.3 Future research

This thesis details only one of the potential SDN applications in the area of SG, which is predominately focused more on the theoretical application of multi-tree BDLDV multicast in SAS. However, papers presented in Chapter 2 indicates that the SDN concept can be beneficial for SG as it allows to substation projects to be transfered directly into the network control systematically. Using reliable and standardized projection and management tools to design and operate SG emphasizes the advantageousness of further automation towards the network control, where the SDN is a viable tool. Eventually, reliable implementation of the control and data plane separation will provide the opportunity to realize ideas that were almost unimaginable in data networks, such as the multi-tree BDLDV multicast. The potential research extensions and recommendations are as follows:

- Extend the proposed multi-tree BDLDV algorithm to multiple sourced multicast, i.e., many-to-many communication.
- Design and implement the multi-tree BDLDV problem using some metaheuristic, preferably using GA, as the results for the LDV problem are encouraging.
- Verify the achieved results from simulations in more complex scenarios, including background traffic typical for SA networks.
- Study the transfer of the multi-tree BDLDV problem to other application areas, for example, distribution of multimedia multicast streams in the closed networks of local providers.

A

Evaluation scenarios

A.1 Isolated evaluation

const_mut1 Evaluated configuration: constant mutation; primary variable: mutpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: Crossover (cxt) = {besttree, one-point}, kp; fixed variables: Number of generations (ng) = 200, Population size (ps) = 200, Tournament size (ts) = 4, cxpb = 0.5, Terminal selection probability (tspb) = 0.5

const_mut2 Evaluated configuration: constant mutation; primary variable: mutpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: mw, li; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

const_mut3 Evaluated configuration: constant mutation; primary variable: mutpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: nd, Mutation-based terminal selection probability (mtsp); fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

const_mut4 Evaluated configuration: constant mutation; primary variable: mutpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: ng = {50, 100, 200, 400}; fixed variables: cxt = besttree, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

const_mut5 Evaluated configuration: constant mutation; primary variable: mutpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: ps = {50, 100, 200, 400}; fixed variables: cxt = besttree, ng = 100, ts = 4, cxpb = 0.5, tspb = 0.5

const_mut6 Evaluated configuration: constant mutation; primary variable: mutpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: ts = {3,5,7,9}; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

const_mut_detail1 Evaluated configuration: constant mutation; primary variable: mutpb= $\langle 0.5, 0.9 \rangle$; secondary configuration combinations: kp, li; fixed variables: cxt = besttree, ng = 400, ps = 400, ts = 9, cxpb = 0.3, tspb = 0.5

const_mut_detail2 Evaluated configuration: constant mutation; primary variable: mutpb= $\langle 0.5, 0.9 \rangle$; secondary configuration combinations: kp, li; fixed variables: cxt = besttree, ng = 100, ps = 200, ts = 9, cxpb = 0.3, tspb = 0.5

intens_mut1 Evaluated configuration: intensified mutation; primary variable: mutpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: cxt = {besttree, onepoint}, kp; fixed variables: ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

intens_mut2 Evaluated configuration: intensified mutation; primary variable: mutpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: mw, li; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

intens_mut3 Evaluated configuration: intensified mutation; primary variable: mutpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: nd, mtsp; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

intens_mut4 Evaluated configuration: intensified mutation; primary variable: mutpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: ng = {50, 100, 200, 400}; fixed variables: cxt = besttree, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

intens_mut5 Evaluated configuration: intensified mutation; primary variable: mutpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: ps = {50, 100, 200, 400}; fixed variables: cxt = besttree, ng = 100, ts = 4, cxpb = 0.5, tspb = 0.5

intens_mut6 Evaluated configuration: intensified mutation; primary variable: mutpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: ts = {3,5,7,9}; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

adapt_mut1 Evaluated configuration: adaptive mutation; primary variable: hdl= $\langle 0.05, 0.25 \rangle$; secondary configuration combinations: cxt = {besttree, onepoint}, kp; fixed variables: ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

adapt_mut2 Evaluated configuration: adaptive mutation; primary variable: hdl= $\langle 0.05, 0.25 \rangle$; secondary configuration combinations: mw, li; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

adapt_mut3 Evaluated configuration: adaptive mutation; primary variable: hdl= $\langle 0.05, 0.25 \rangle$; secondary configuration combinations: nd, mtsp; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

adapt_mut4 Evaluated configuration: adaptive mutation; primary variable: hdl= $\langle 0.05, 0.25 \rangle$; secondary configuration combinations: ng = {50, 100, 200, 400}; fixed variables: cxt = besttree, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

adapt_mut5 Evaluated configuration: adaptive mutation; primary variable: hdl= $\langle 0.05, 0.25 \rangle$; secondary configuration combinations: ps = {50, 100, 200, 400}; fixed variables: cxt = besttree, ng = 100, ts = 4, cxpb = 0.5, tspb = 0.5

adapt_mut6 Evaluated configuration: adaptive mutation; primary variable: hdl= $\langle 0.05, 0.25 \rangle$; secondary configuration combinations: ts = {3,5,7,9}; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5

adapt_mut_detail1 Evaluated configuration: adaptive mutation; primary variable: hdl= $\langle 0.08, 0.12 \rangle$; secondary configuration combinations: kp, li; fixed variables: cxt = besttree, ng = 400, ps = 400, ts = 9, cxpb = 0.3, tspb = 0.5

adapt_mut_detail2 Evaluated configuration: adaptive mutation; primary variable: hdl= $\langle 0.08, 0.12 \rangle$; secondary configuration combinations: kp, li; fixed variables: cxt = besttree, ng = 100, ps = 200, ts = 9, cxpb = 0.3, tspb = 0.5

adapt_mut_detail3 Evaluated configuration: adaptive mutation; primary variable: hdl= $\langle 0.1, 0.2 \rangle$; secondary configuration combinations: kp, li; fixed variables: cxt = besttree, ng = 100, ps = 200, ts = 9, cxpb = 0.3, tspb = 0.5

besttree_cxt1 Evaluated configuration: besttree crossover; primary variable: cxpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: Mutation (mut) = {constant, intensified, adaptive}; fixed variables: ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5, mutpb = 0.5, hdl = 0.1

besttree_cxt2 Evaluated configuration: besttree crossover; primary variable: cxpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: mw, li; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5, mutpb = 0.5, mut = const

besttree_cxt3 Evaluated configuration: besttree crossover; primary variable: cxpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: nd, kp; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5, mutpb = 0.5, mut = const

besttree_cxt4 Evaluated configuration: besttree crossover; primary variable: cxpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: ng = {50, 100, 200, 400}; fixed variables: cxt = besttree, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5, mutpb = 0.5, mut = const

besttree_cxt5 Evaluated configuration: besttree crossover; primary variable: cxpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: ps = {50, 100, 200, 400}; fixed variables: cxt = besttree, ng = 100, ts = 4, cxpb = 0.5, tspb = 0.5, mutpb = 0.5, mut = const

besttree_cxt6 Evaluated configuration: besttree crossover; primary variable: cxpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: ts = {3,5,7,9}; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5, mutpb = 0.5, mut = const

onepoint_cxt1 Evaluated configuration: onepoint crossover; primary variable: cxpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: mut = {constant, intensified, adaptive}; fixed variables: ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5, mutpb = 0.5, hdl = 0.1

onepoint_cxt2 Evaluated configuration: onepoint crossover; primary variable: cxpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: mw, li; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5, mutpb = 0.5, mut = const

onepoint_cxt3 Evaluated configuration: onepoint crossover; primary variable: cxpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: nd, kp; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5, mutpb = 0.5, mut = const

onepoint_cxt4 Evaluated configuration: onepoint crossover; primary variable: cxpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: ng = {50, 100, 200, 400}; fixed variables: cxt = besttree, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5, mutpb = 0.5, mut = const

onepoint_cxt5 Evaluated configuration: onepoint crossover; primary variable: cxpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: ps = {50, 100, 200, 400}; fixed variables: cxt = besttree, ng = 100, ts = 4, cxpb = 0.5, tspb = 0.5, mutpb = 0.5, mut = const

onepoint_cxt6 Evaluated configuration: onepoint crossover; primary variable: cxpb= $\langle 0.1, 0.9 \rangle$; secondary configuration combinations: ts = {3,5,7,9}; fixed variables: cxt = besttree, ng = 200, ps = 200, ts = 4, cxpb = 0.5, tspb = 0.5, mutpb = 0.5, mut = const

A.2 Hyperparameter search domains

The search domains listed below describe numerical ranges and set of features the hyperparameter optimization algorithm takes into account when seeking the best configuration combination in the parameter search space. Features without defined sets are of Boolean type, and can be only activated or deactivated. Following variable ranges are shared across all scenarios: ts = $\langle 3, 9 \rangle$, ps = $\langle 50, 400 \rangle$, ng = $\langle 50, 300 \rangle$, cxpb = $\langle 0, 1 \rangle$, mutpb = $\langle 0, 1 \rangle$.

const_mut cxt = {besttree, onepoint}, mut = {constant}, mw, kp, li, nd

adapt_mut cxt = {besttree, onepoint}, mut = {adaptive}, mw, kp, li, hdl = $\langle 0.01, 0.25 \rangle$

adapt_mut2 cxt = {besttree, onepoint}, mut = {adaptive}, mw, kp, li, mtsp, hdl = $\langle 0.01, 0.25 \rangle$

besttree_cxt cxt = {besttree}, mut = {constant, intensified}, mw, kp, li

onepoint_cxt cxt = {onepoint}, mut = {constant, intensified}, mw, kp, li

besttree_cxt2 cxt = {besttree}, mut = {adaptive}, mw, kp, li, hdl = $\langle 0.01, 0.25 \rangle$

onepoint_cxt2 cxt = {onepoint}, mut = {adaptive}, mw, kp, li, hdl = $\langle 0.01, 0.25 \rangle$

Bibliography

[1] *ISO/IEC/IEEE 8802-3:2017, Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Standard for Ethernet*, International Organization for Standardization, 2017.

[2] H. Georg, N. Dorsch, M. Putzke, and C. Wietfeld, "Performance evaluation of time-critical communication networks for smart grids based on iec 61850", in *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, 2013, pp. 43–48. DOI: 10.1109/INFCOMW.2013.6562906.

[3] E. Molina, E. Jacob, J. Matias, N. Moreira, and A. Astarloa, "Using software defined networking to manage and control {iec} 61850-based systems", *Computers & Electrical Engineering*, vol. 43, pp. 142 –154, 2015, ISSN: 0045-7906. DOI: http://dx.doi.org/10.1016/j.compeleceng.2014.10.016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790614002626.

[4] X. Dong, H. Lin, R. Tan, R. K. Iyer, and Z. Kalbarczyk, "Software-defined networking for smart grid resilience: Opportunities and challenges", in *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*, ser. CPSS '15, Singapore, Republic of Singapore: ACM, 2015, pp. 61–68, ISBN: 978-1-4503-3448-8. DOI: 10.1145/2732198.2732203. [Online]. Available: http://doi.acm.org/10.1145/2732198.2732203.

[5] *IEC TS 61850-2:2003, Communication networks and systems in substations - Part 2: Glossary*, International Electrotechnical Commission, 2003.

[6] C. Brunner, "Implementation guideline for digital interace to instrument transformers using iec 61850-9-2", UCA International Users Group, Tech. Rep., 2007.

[7] ABB. (2010). Glossary of technical terms, ABB, [Online]. Available: http://bit.ly/2oMsf8K.

[8] T. Skeie, S. Johannessen, and C. Brunner, "Ethernet in substation automation", *Control Systems, IEEE*, vol. 22, no. 3, pp. 43–51, 2002, ISSN: 1066-033X. DOI: 10.1109/MCS.2002.1003998.

[9] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, *Network configuration protocol (netconf)*, RFC6241, 2011. [Online]. Available: http://tools.ietf.org/rfc/rfc6241.txt.

[10] H. J. Prömel and A. Steger, *The Steiner Tree Problem: A Tour through Graphs, Algorithms, and Complexity (Advanced Lectures in Mathematics)*. Vieweg+Teubner Verlag, 2002, ISBN: 3528067624.

[11] *IEC 61850-5:2013, Communication networks and systems for power utility automation - Part 5: Communication requirements for functions and device models*, International Electrotechnical Commission, 2011.

[12] J. Li, Q. Huang, F. kai Hu, and S. Jing, "Performance testing on GOOSE and MSV transmission in one network", *Energy Procedia*, vol. 12, no. 0, pp. 185 –191, 2011, The Proceedings of International Conference on Smart Grid and Clean Energy Technologies (ICSGCE 2011, ISSN: 1876-6102. DOI: http://dx.doi.org/10.1016/j.egypro.2011.10.026. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1876610211018522.

[13] D. M. E. Ingram, P. Schaub, R. R. Taylor, and D. A. Campbell, "Performance analysis of iec 61850 sampled value process bus networks", *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1445–1454, 2013, ISSN: 1551-3203. DOI: 10.1109/TII.2012.2228874.

[14] W. Fenner, *Internet group management protocol, version 2*, RFC2236, 1997. [Online]. Available: http://tools.ietf.org/rfc/rfc2236.txt.

[15] D. Ingram, P. Schaub, and D. Campbell, "Multicast traffic filtering for sampled value process bus networks", in *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, 2011, pp. 4710–4715. DOI: 10.1109/IECON.2011.6120087.

[16] "IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks", *IEEE Std 802.1Q-2014 (Revision of IEEE Std 802.1Q-2011)*, pp. 1–1832, 2014. DOI: 10.1109/IEEESTD.2014.6991462.

[17] IEEE, *IEEE Standard for Local and metropolitan area networks–Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks*, IEEE, 2011.

[18] *IEC 62439-3:2016, Industrial communication networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR)*, International Electrotechnical Commission, 2016.

[19] R. Novak, J. Rugelj, and G. Kandus, *Steiner tree based distributed multicast routing in networks*. Springer, 2001.

[20] J. Moy, *Multicast extensions to ospf*, RFC1584, 1994. [Online]. Available: http://tools.ietf.org/rfc/rfc1584.txt.

[21] D. Waitzman, C. Partridge, and S. Deering, *Distance vector multicast routing protocol*, RFC1075, 1988. [Online]. Available: http://tools.ietf.org/rfc/rfc1075.txt.

[22] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, *Protocol independent multicast - sparse mode (pim-sm): Protocol specification (revised)*, RFC4601, 2006. [Online]. Available: http://tools.ietf.org/rfc/rfc4601.txt.

[23] A. Adams, J. Nicholas, and W. Siadak, *Protocol independent multicast - dense mode (pim-dm): Protocol specification (revised)*, RFC3973, 2005. [Online]. Available: http://tools.ietf.org/rfc/rfc3973.txt.

[24] A. Ballardie, *Core based trees (cbt version 2) multicast routing – protocol specification –*, RFC2189, 1997. [Online]. Available: http://tools.ietf.org/rfc/rfc2189.txt.

[25] "IEEE Standard for Local and metropolitan area networks–Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks–Amendment 20: Shortest Path Bridging", *IEEE Std 802.1aq-2012 (Amendment to IEEE Std 802.1Q-2011 as amended by IEEE Std 802.1Qbe-2011, IEEE Std 802.1Qbc-2011, IEEE Std 802.1Qbb-2011, IEEE Std 802.1Qaz-2011, and IEEE Std 802.1Qbf-2011)*, pp. 1–340, 2012. DOI: 10.1109/IEEESTD.2012.6231597.

[26] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on smart grid communication infrastructures: Motivations, requirements and challenges", *Communications Surveys Tutorials, IEEE*, vol. 15, no. 1, pp. 5–20, 2013, ISSN: 1553-877X. DOI: 10.1109/SURV.2012.021312.00034.

[27] "Regulation (Eu) No 1025/2012 of the European Parliament and of The Council of 25 October 2012 on European standardization, amending Council Directives 89/686/EEC and 93/15/EEC and Directives 94/9/EC, 94/25/EC, 95/16/EC, 97/23/EC, 98/34/EC, 2004/22/EC, 2007/23/EC, 2009/23/EC and 2009/105/EC of the European Parliament and of the Council and repealing Council Decision 87/95/EEC and Decision No 1673/2006/EC of the European Parliament and of the Council", *Official Journal of the European Union*, vol. L 316/12, 2012.

[28] EUROPEAN COMMISSION, DIRECTORATE-GENERAL FOR ENERGY, *M/490, Smart Grid Mandate, Standardization Mandate to European Standardisation Organisations (ESOs) to support European Smart Grid deployment*, 2011. [Online]. Available: https://bit.ly/2CWUjhq.

[29] Smart Grid and Cyber - Physical Systems Program Office and Energy and Environment Division, Engineering Laboratory, "NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0", NIST, Tech. Rep., 2014. [Online]. Available: https://bit.ly/2KAIW2h.

[30] EUROPEAN COMMISSION, ENTERPRISE AND INDUSTRY DIRECTORATE-GENERAL, *M/468, STANDARDISATION MANDATE TO CEN, CENELEC AND ETSI CONCERNING THE CHARGING OF ELECTRIC VEHICLES*, 2010. [Online]. Available: https://bit.ly/31CH2DR.

[31] EUROPEAN COMMISION ENTERPRISE AND INDUSTRY DIRECTORATE-GENERAL, *M/441, Standardisation mandate to CEN, CENELEC and ETSI in the field of measuring instruments for the development of an open architecture for utility meters involving communication protocols enabling interoperability*, 2009. [Online]. Available: https://bit.ly/2YPdsOw.

[32] CEN-CENELEC-ETSI Coordination Group on Smart Energy Grids (CG-SEG), "SEGCG/M490/G_Smart Grid Set of Standards, Version 4.1", CCMC, Tech. Rep., 2017. [Online]. Available: https://bit.ly/2YVmY2R.

[33] CEN-CENELEC-ETSI Smart Grid Coordination Group, "CEN-CENELEC-ETSI Smart Grid Coordination Group Smart Grid Reference Architecture", Tech. Rep., 2012. [Online]. Available: https://bit.ly/2nEbPy1.

[34] *IEC 61850-8-1:2011, Communication networks and systems for power utility automation - Part 8-1: Specific communication service mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3*, International Electrotechnical Commission, 2011.

[35] Z. XiCai, W. ShuChao, X. Lei, and F. YaDong, "Practice and trend of dsas in china", in *Advanced Power System Automation and Protection (APAP), 2011 International Conference on*, vol. 3, 2011, pp. 1762–1766. DOI: 10.1109/APAP.2011.6180836.

[36] C. Fan, "Data acquisition applications", in, Z. Karakehayov, Ed. InTech, 2012, ch. Chapter 6 - The Data Acquisition in Smart Substation of China, pp. 123–164. DOI: http://dx.doi.org/10.5772/47853. [Online]. Available: http://www.intechopen.com/books/export/citation/BibTex/data-acquisition-applications/the-data-acquisition-in-smart-substation-of-china.

[37] *IEC 61850-7-2:2010, Communication networks and systems for power utility automation - Part 7-2: Basic information and communication structure - Abstract communication service interface (ACSI)*, International Electrotechnical Commission, 2010.

[38] *IEC 61850-9-2:2011, Communication networks and systems for power utility automation - Part 9-2: Specific communication service mapping (SCSM) - Sampled values over ISO/IEC 8802-3*, International Electrotechnical Commission, 2011.

[39] *ISO 9506-1:2003, Industrial automation systems – Manufacturing Message Specification – Part 1: Service definition*, International Organization for Standardization, 2003.

[40] *ISO 9506-2:2003, Industrial automation systems – Manufacturing Message Specification – Part 2: Protocol specification*, International Organization for Standardization, 2003.

[41] "IEEE Standard for a precision clock synchronization protocol for networked measurement and control systems", *IEC 61588:2009(E)*, pp. C1–274, 2009. DOI: 10.1109/IEEESTD.2009.4839002.

[42] "Ieee standard for a precision clock synchronization protocol for networked measurement and control systems", *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pp. 1–300, 2008. DOI: 10.1109/IEEESTD.2008.4579760.

[43] "IEC/IEEE International Standard - Communication networks and systems for power utility automation Part 9-3: Precision time protocol profile for power utility automation", *IEC/IEEE 61850-9-3 Edition 1.0 2016-05*, pp. 1–18, 2016. DOI: 10.1109/IEEESTD.2016.7479438.

[44] B. S. K. Jr., "A layman's guide to a subset of asn.1, ber, and der", RSA Laboratories, Tech. Rep., 1993. [Online]. Available: ftp://ftp.rsasecurity.com/pub/pkcs/ascii/layman.asc.

[45] *X.690 : Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*, 2016.

[46] F. Becker, S. Nohe, and A. Echeverria, *Designing non-deterministic pac systems to meet deterministic requirements*, NYPA, USA, 2015. [Online]. Available: https://bit.ly/305lQWO.

[47] *IEC 61869-9:2016, Instrument transformers - Part 9: Digital interface for instrument transformers*, International Electrotechnical Commission, 2016.

[48] Ruggedcom Inc., "eRSTP - enhanced Rapid Spanning Tree Protocol", Tech. Rep. [Online]. Available: http://www.ruggedcom.com/products/eRSTP/.

[49] P. Ashwood-Smith, "Shortest path bridging ieee 802.1aq overview", APRICOT/Hong Kong, 2011, [Online]. Available: https://bit.ly/2MiwOVr.

[50] D. E. E. 3rd, T. Senevirathne, A. Ghanwani, D. Dutt, and A. Banerjee, *Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS*, RFC 7176, 2014. DOI: 10.17487/RFC7176. [Online]. Available: https://rfc-editor.org/rfc/rfc7176.txt.

[51] P. Ferrari, A. Flammini, S. Rinaldi, G. Prytz, and R. Hussain, "Multipath redundancy for industrial networks using ieee 802.1aq shortest path bridging", in *2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014)*, 2014, pp. 1–10. DOI: 10.1109/WFCS.2014.6837598.

[52] D. E. E. 3rd, D. G. Dutt, S. Gai, R. Perlman, and A. Ghanwani, *Routing Bridges (RBridges): Base Protocol Specification*, RFC 6325, 2011. DOI: 10.17487/RFC6325. [Online]. Available: https://rfc-editor.org/rfc/rfc6325.txt.

[53] S. Zuboff, *Be the friction - our response to the new lords of the ring*, F. A. Zeitung, Ed., 2013. [Online]. Available: https://bit.ly/30awLyA.

[54] S. Sezer, S. Scott-Hayward, P. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for sdn? implementation challenges for software-defined networks", *Communications Magazine, IEEE*, vol. 51, no. 7, pp. 36–43, 2013, ISSN: 0163-6804. DOI: 10.1109/MCOM.2013.6553676.

[55] P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. C. Liaw, T. Lyon, and G. Minshall, *Ipsilon's general switch management protocol specification version 1.1*, RFC1987, 1996. [Online]. Available: http://tools.ietf.org/rfc/rfc1987.txt.

[56] J. Van der Merwe, S. Rooney, I Leslie, and S. Crosby, "The tempest-a practical framework for network programmability", *Network, IEEE*, vol. 12, no. 3, pp. 20–28, 1998, ISSN: 0890-8044. DOI: 10.1109/65.690958.

[57] L. Yang, R. Dantu, T. Anderson, and R. Gopal, *Forwarding and control element separation (forces) framework*, RFC3746, 2004. [Online]. Available: http://tools.ietf.org/rfc/rfc3746.txt.

[58] A. Farrel, J.-P. Vasseur, and J. Ash, *A path computation element (pce)-based architecture*, RFC4655, 2006. [Online]. Available: http://tools.ietf.org/rfc/rfc4655.txt.

[59] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking control of the enterprise", *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 1–12, Aug. 2007, ISSN: 0146-4833. DOI: 10.1145/1282427.1282382. [Online]. Available: http://doi.acm.org/10.1145/1282427.1282382.

[60] O. N. Foundation, *Openflow switch specification 1.5.1*, Open Networking Foundation, 2015. [Online]. Available: https://bit.ly/2KJ9Jbm.

[61] Cisco, "One platform kit (onepk) for developers", Cisco, Tech. Rep., 2014. [Online]. Available: http://www.cisco.com/c/dam/en/us/products/collateral/ios-nx-os-software/at_a_glance_c45-708540.pdf.

[62] ONF, "Software-defined networking: The new norm for networks", Open Networking Foundation, Tech. Rep., 2012. [Online]. Available: https://bit.ly/1ma4Pii.

[63] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, *Requirements for traffic engineering over mpls*, RFC2702, 1999. [Online]. Available: `http://tools.ietf.org/rfc/rfc2702.txt`.

[64] J. Vasseur and J. L. Roux, *Path computation element (pce) communication protocol (pcep)*, RFC5440, 2009. [Online]. Available: `http://tools.ietf.org/rfc/rfc5440.txt`.

[65] Y. Rekhter, T. Li, and S. Hares, *A border gateway protocol 4 (bgp-4)*, RFC4271, 2006. [Online]. Available: `http://tools.ietf.org/rfc/rfc4271.txt`.

[66] P. Saint-Andre, *Extensible messaging and presence protocol (xmpp): Core*, RFC6120, 2011. [Online]. Available: `http://tools.ietf.org/rfc/rfc6120.txt`.

[67] S. Mackie, L. Fang, N. Sheth, M. Napierala, and N. Bitar, "BGP-Signaled End-System IP/VPNs", IETF Secretariat, Internet-Draft draft-ietf-l3vpn-end-system-06, 2016. [Online]. Available: `http://www.ietf.org/internet-drafts/draft-ietf-l3vpn-end-system-06.txt`.

[68] R. Ahmed and R. Boutaba, "Design considerations for managing wide area software defined networks", *Communications Magazine, IEEE*, vol. 52, no. 7, pp. 116–123, 2014, ISSN: 0163-6804. DOI: `10.1109/MCOM.2014.6852092`.

[69] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, *et al.*, "Onix: A distributed control platform for large-scale production networks.", in *OSDI*, vol. 10, 2010, pp. 1–6.

[70] A. Tootoonchian and Y. Ganjali, "Hyperflow: A distributed control plane for openflow", in *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, ser. INM/WREN'10, San Jose, CA: USENIX Association, 2010, pp. 3–3. [Online]. Available: `http://dl.acm.org/citation.cfm?id=1863133.1863136`.

[71] S. Hassas Yeganeh and Y. Ganjali, "Kandoo: A framework for efficient and scalable offloading of control applications", in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12, Helsinki, Finland: ACM, 2012, pp. 19–24, ISBN: 978-1-4503-1477-0. DOI: `10.1145/2342441.2342446`. [Online]. Available: `http://doi.acm.org/10.1145/2342441.2342446`.

[72] *Opendaylight controller:programmer guide:clustering*, OpenDaylight Project. [Online]. Available: `https://bit.ly/2Z5FNiP`.

[73] A. Koshibe and E. Olkhovskaya, *Cluster coordination*, ONOS, 2016. [Online]. Available: `https://wiki.onosproject.org/display/ONOS/Cluster+Coordination`.

[74] H. Yin, H. Xie, T. Tsou, D. Lopez, P. Aranda, and R. Sidi, "Sdni: A message exchange protocol for software defined networks (sdns) across multiple domains", IETF Secretariat, Internet-Draft draft-yin-sdn-sdni-00, 2012. [Online]. Available: `http://www.ietf.org/internet-drafts/draft-yin-sdn-sdni-00.txt`.

[75] ONF, "Openflow switch specification 1.0.0", Open Networking Foundation, Tech. Rep., 2009. [Online]. Available: `https://bit.ly/1hOA3Os`.

[76] O. N. Foundation, *MPLS-TP OpenFlow Protocol Extensions for SPTN*, Open Networking Foundation, 2017. [Online]. Available: `http://bit.ly/2DnPgTW`.

[77] T. Hegr, L. Bohac, V. Uhlir, and P. Chlumsky, "OpenFlow deployment and concept analysis", *Advances in Electrical and Electronic Engineering*, vol. 11, no. 5, pp. 327–335, 2013, ISSN: 1336-1376. DOI: 10.15598/aeee.v11i5.884.

[78] A. Patwary, B. Geuskens, and S. Lu, "Low-power ternary content addressable memory (tcam) array for network applications", in *Communications, Circuits and Systems, 2009. ICCCAS 2009. International Conference on*, 2009, pp. 322–325. DOI: 10.1109/ICCCAS.2009.5250516.

[79] D. Warren, *Switch with adaptive address lookup hashing scheme*, USPatent 6690667, 2004. [Online]. Available: http://www.google.com/patents/US6690667.

[80] R. Diestel, *Graph Theory (Graduate Texts in Mathematics)*. Springer, 2010, ISBN: 3642142788.

[81] B. Bollobás, "Random graphs", English, in *Modern Graph Theory*, ser. Graduate Texts in Mathematics, vol. 184, Springer New York, 1998, pp. 215–252, ISBN: 978-0-387-98488-9. DOI: 10.1007/978-1-4612-0619-4_7. [Online]. Available: http://dx.doi.org/10.1007/978-1-4612-0619-4_7.

[82] M. A. Porter, "Small-world network", *Scholarpedia*, vol. 7, no. 2, p. 1739, 2012, revision #170493. DOI: 10.4249/scholarpedia.1739.

[83] D. J. Watts and S. H. Strogatz, "Collective dynamics of a small-world networks", *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[84] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks", *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[85] S. Dorogovtsev and J. Mendes, "Evolution of networks", *ADVANCES IN PHYSICS*, vol. 51, no. 4, 1079–1187, 2002, ISSN: 0001-8732. DOI: 10.1080/00018730110112519.

[86] M. Manzano, E. Calle, and D. Harle, "Quantitative and qualitative network robustness analysis under different multiple failure scenarios", in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 3rd International Congress on*, 2011, pp. 1–7.

[87] A. Sydney, C. M. Scoglio, P. Schumm, and R. E. Kooij, "Elasticity: Topological characterization of robustness in complex networks", *CoRR*, vol. abs/0811.4040, 2008.

[88] T. Hegr and L. Bohac, "Impact of nodal centrality measures to robustness in software-defined networking", *Advances in Electrical and Electronic Engineering*, vol. 12, no. 4, pp. 252–259, 2014, ISSN: 1336-1376. DOI: 10.15598/aeee.v12i4.1208.

[89] S. Neumayer and E. Modiano, "Network reliability with geographically correlated failures", in *INFOCOM, 2010 Proceedings IEEE*, IEEE, 2010, pp. 1–9.

[90] C. Clos, "A study of non-blocking switching networks", *Bell System Technical Journal, The*, vol. 32, no. 2, pp. 406–424, 1953, ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1953.tb01433.x.

[91] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture", in *ACM SIGCOMM Computer Communication Review*, ACM, vol. 38, 2008, pp. 63–74.

[92] Cisco, "Cisco data center infrastructure 2.5 design guide", Cisco Systems, Inc., Tech. Rep., 2011. [Online]. Available: http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/DC_Infra2_5/DCI_SRND_2_5a_book.pdf.

[93] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: Measurement, analysis, and implications", *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 350–361, Aug. 2011, ISSN: 0146-4833. DOI: 10.1145/2043164.2018477. [Online]. Available: http://doi.acm.org/10.1145/2043164.2018477.

[94] H. F. Salama, "Multicast routing for real-time communication of high-speed networks", AAI9710761, PhD thesis, 1996, ISBN: 0-591-18848-1.

[95] R. Karp, "Reducibility among combinatorial problems", English, in *Complexity of Computer Computations*, ser. The IBM Research Symposia Series, R. Miller, J. Thatcher, and J. Bohlinger, Eds., Springer US, 1972, pp. 85–103, ISBN: 978-1-4684-2003-6. DOI: 10.1007/978-1-4684-2001-2_9. [Online]. Available: http://dx.doi.org/10.1007/978-1-4684-2001-2_9.

[96] G. N. Rouskas and I. Baldine, "Multicast routing with end-to-end delay and delay variation constraints", *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 346–356, 1997, ISSN: 0733-8716. DOI: 10.1109/49.564133.

[97] S. Voß, "Steiner's problem in graphs: Heuristic methods", *Discrete Applied Mathematics*, vol. 40, no. 1, pp. 45 –72, 1992, ISSN: 0166-218X. DOI: http://dx.doi.org/10.1016/0166-218X(92)90021-2. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0166218X92900212.

[98] M. Moh and B. Nguyen, "Qos-guaranteed one-to-many and many-to-many multicast routing", *Computer Communications*, vol. 26, no. 7, pp. 652 –669, 2003, Advances in Computer Communications and Networks: Algorithms and Applications, ISSN: 0140-3664. DOI: http://dx.doi.org/10.1016/S0140-3664(02)00198-6. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366402001986.

[99] R. C. Prim, "Shortest connection networks and some generalizations", *The Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957, ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1957.tb01515.x.

[100] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees", English, *Acta Informatica*, vol. 15, no. 2, pp. 141–145, 1981, ISSN: 0001-5903. DOI: 10.1007/BF00288961. [Online]. Available: http://dx.doi.org/10.1007/BF00288961.

[101] M. R. Kabat, M. K. Patel, and C. R. Tripathy, "A heuristic algorithm for delay delay-variation bounded least cost multicast routing", in *Advance Computing Conference (IACC), 2010 IEEE 2nd International*, 2010, pp. 261–266. DOI: 10.1109/IADCC.2010.5423000.

[102] M. F. Mokbel, W. A. Elhaweet, and M. N. Elderini, "An efficient algorithm for shortest path multicast routing under delay and delay variation constraints", in *Proceedings of the Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, Citeseer, 2000, pp. 190–196.

[103] M. Kim, Y.-C. Bang, and H. Choo, "High performance computing and communications: Second international conference, hpcc 2006, munich, germany, september 13-15, 2006. proceedings", in, M. Gerndt and D. Kranzlmüller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, ch. On Multicasting Steiner Trees for Delay and Delay Variation Constraints, pp. 447–456, ISBN: 978-3-540-39372-6. DOI: 10.1007/11847366_46. [Online]. Available: http://dx.doi.org/10.1007/11847366_46.

[104] H. Takahashi and A. Matsuyama, "An approximate solution for the steiner problem in graphs", *Mathematica Japonica*, vol. 6, no. 24, pp. 573–577, 1980.

[105] R. Widyono *et al.*, *The design and evaluation of routing algorithms for real-time channels*. 1994.

[106] Q. Zhu, M. Parsa, and J. Garcia-Luna-Aceves, "A source-based algorithm for delay-constrained minimum-cost multicasting", in *INFOCOM '95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People. Proceedings. IEEE*, 1995, 377–385 vol.1. DOI: 10.1109/INFCOM.1995.515898.

[107] V. Kompella, J. Pasquale, and G. Polyzos, "Multicast routing for multimedia communication", *Networking, IEEE/ACM Transactions on*, vol. 1, no. 3, pp. 286–292, 1993, ISSN: 1063-6692. DOI: 10.1109/90.234851.

[108] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications", *Selected Areas in Communications, IEEE Journal on*, vol. 14, no. 7, pp. 1228–1234, 1996, ISSN: 0733-8716. DOI: 10.1109/49.536364.

[109] S. Verma, R. K. Pankaj, and A. Leon-Garcia, "Qos based multicast routing algorithms for real time applications", *Performance Evaluation*, vol. 34, no. 4, pp. 273 –294, 1998, ISSN: 0166-5316. DOI: http://dx.doi.org/10.1016/S0166-5316(98)00041-8. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166531698000418.

[110] F. Glover and K. Sörensen, "Metaheuristics", *Scholarpedia*, vol. 10, no. 4, p. 6532, 2015, revision #149834. DOI: 10.4249/scholarpedia.6532.

[111] Z. Kun, W. Heng, and L. Feng-Yu, "Distributed multicast routing for delay and delay variation-bounded steiner tree using simulated annealing", *Computer Communications*, vol. 28, no. 11, pp. 1356 –1370, 2005, ISSN: 0140-3664. DOI: http://dx.doi.org/10.1016/j.comcom.2004.12.003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366404003950.

[112] C. C. Ribeiro and M. C. De Souza, "Tabu search for the steiner problem in graphs", *Networks*, vol. 36, no. 2, pp. 138–146, 2000.

[113] N. Skorin-Kapov and M. Kos, "A grasp heuristic for the delay-constrained multicast routing problem", *Telecommunication Systems*, vol. 32, no. 1, pp. 55–69, 2006.

[114] M. Hamdan and M. El-Hawary, "A novel genetic algorithm searching approach for dynamic constrained multicast routing", in *Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on*, vol. 2, 2003, 1127–1130 vol.2. DOI: 10.1109/CCECE.2003.1226095.

[115] M. Hamdan and M. E. El-Hawary, "Multicast routing with delay and delay variation constraints using genetic algorithm", in *Electrical and Computer Engineering, 2004. Canadian Conference on*, vol. 4, 2004, 2363–2366 Vol.4. DOI: 10.1109/CCECE.2004.1347721.

[116] M. Terzian, "On near optimal time and dynamic delay and delay variation multicast algorithms", PhD thesis, Concordia University, Department of Computer Science and Software Engineering, Montreal, Quebec, Canada, 2017.

[117] Y. Xu, R. Qu, and R. Li, "A simulated annealing based genetic local search algorithm for multi-objective multicast routing problems", *Annals of Operations Research*, vol. 206, no. 1, pp. 527–555, 2013, ISSN: 1572-9338. DOI: 10.1007/s10479-013-1322-7. [Online]. Available: http://dx.doi.org/10.1007/s10479-013-1322-7.

[118] H.-C. Lin, T.-M. Lin, and C.-F. Wu, "Constructing application-layer multicast trees for minimum-delay message distribution", *Information Sciences*, vol. 279, pp. 433–445, 2014, ISSN: 0020-0255. DOI: http://dx.doi.org/10.1016/j.ins.2014.03.130. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025514004277.

[119] J. W. Park, C. K. Hwang, and Y. W. Lee, "New ilp formulations for multicast routing in sparse-splitting optical networks", in *2012 8th international conference on network and service management (cnsm) and 2012 workshop on systems virtualiztion management (svm)*, 2012, pp. 238–241.

[120] G. Giambene, *Queuing Theory and Telecommunications: Networks and Applications*. Springer, 2005, ISBN: 0387240659.

[121] "Latency on a switched ethernet network", Siemens AG, Tech. Rep., 2014. [Online]. Available: https://sie.ag/2Z6b4Cq.

[122] T. Hégr, L. Boháč, Z. Kocur, M. Vozňák, and P. Chlumský, "Methodology of the direct measurement of the switching latency", English, *Przeglad Elektrotechniczny*, vol. 89, no. 7/2013, pp. 59–63, 2013, ISSN: 0033-2097. [Online]. Available: http://pe.org.pl/articles/2013/7/13.pdf.

[123] T. Hegr, M. Voznak, M. Kozak, and L. Bohac, "Measurement of Switching Latency in High Data Rate Ethernet Networks", *Elektronika IR Elektrotechnika*, vol. 21, no. 3, 73–78, 2015, ISSN: 1392-1215. DOI: 10.5755/j01.eee.21.3.10445.

[124] J. Loeser and H. Haertig, "Low-latency hard real-time communication over switched ethernet", in *Real-Time Systems, 2004. ECRTS 2004. Proceedings. 16th Euromicro Conference on*, 2004, pp. 13 –22. DOI: 10.1109/EMRTS.2004.1310992.

[125] M Pravda, P Lafata, and J Vodrážka, "Precision clock synchronization protocol and its implementation into laboratory ethernet network", in *33rd International Conference on Telecommunication and Signal Processing. Vienna, Austria*, 2010, pp. 286–291, ISBN: 978-963-88981-0-4.

[126] A. Poursepanj. (2003). Benchmarks rate switch-fabric performance, CommsDesign, [Online]. Available: http://m.eet.com/media/1095854/feat1-dec03.pdf.

[127] *IEEE Standard for Ethernet - Section 1*, IEEE, 2012. DOI: 10.1109/IEEESTD.2012.6419735.

[128] S. Bradner, *Benchmarking terminology for network intercon. devices*, RFC1242, 1991. [Online]. Available: `http://tools.ietf.org/rfc/rfc1242.txt`.

[129] S. Bradner and J. McQuaid, *Benchmarking methodology for network interconnect devices*, RFC2544, 1999. [Online]. Available: `http://tools.ietf.org/rfc/rfc2544.txt`.

[130] JCGM, "Jcgm 100: Evaluation of measurement data - guide to the expression of uncertainty in measurement", Joint Committee for Guides in Metrology, Tech. Rep., 2008. [Online]. Available: `http://goo.gl/ryF5ka`.

[131] H Chao, *High performance switches and routers*. Hoboken, N.J: Wiley-Interscience, 2007, ISBN: 978-0-470-05367-6.

[132] M. Havlan. (2007). Aktivní diferenciální sondy. Czech, [Online]. Available: `http://access.feld.cvut.cz/view.php?cisloclanku=2007110003`.

[133] R. Mandeville and J. Perser, *Benchmarking methodology for lan switching devices*, RFC2889, 2000. [Online]. Available: `http://tools.ietf.org/rfc/rfc2889.txt`.

[134] H. Grecco, *Python VISA library*, 2013. [Online]. Available: `https://github.com/hgrecco/pyvisa`.

[135] Floodlight, *Static Flow Pusher API*, 2013. [Online]. Available: `http://goo.gl/fMDfaI`.

[136] A. J. Conejo, *Decomposition Techniques in Mathematical Programming: Engineering and Science Applications*. Springer, 2006, ISBN: 3540276858. [Online]. Available: `https://www.xarg.org/ref/a/3540276858/`.

[137] E. Oki, *Linear Programming and Algorithms for Communication Networks: A Practical Guide to Network Design, Control, and Management*. CRC Press, 2016, ISBN: 1138034096. [Online]. Available: `https://bit.ly/2Pzlmnj`.

[138] M. S. Bazaraa, *Linear Programming and Net. Flows*. Wiley, 2009, ISBN: 0470462728. [Online]. Available: `https://www.xarg.org/ref/a/0470462728/`.

[139] G. B. Dantzig, "Application of the simplex method to a transportation problem", in *Activity Analysis of Production and Allocation*, ser. Cowles Commission Monograph No. 13, John Wiley & Sons Inc., New York; Chapman & Hall Ltd., London, 1951, pp. 359–373.

[140] R. J. Vanderbei, *Linear Programming: Foundations and Extensions (International Series in Operations Research Management Science, 37.)* Springer, 2001, ISBN: 0792373421. [Online]. Available: `https://www.xarg.org/ref/a/0792373421/`.

[141] E. W. Dijkstra, "A note on two problems in connexion with graphs", *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959, ISSN: 0945-3245. DOI: `10.1007/BF01386390`. [Online]. Available: `https://doi.org/10.1007/BF01386390`.

[142] A. Schrijver, "On the history of the transportation and maximum flow problems", *Mathematical Programming*, vol. 91, no. 3, pp. 437–445, 2002, ISSN: 1436-4646. DOI: `10.1007/s101070100259`. [Online]. Available: `https://doi.org/10.1007/s101070100259`.

[143] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network", *Canadian Journal of Mathematics*, vol. 8, 399–404, 1956. DOI: `10.4153/CJM-1956-045-5`.

[144] T. Hegr, M. Kozak, and L. Bohac, "On Application of Least-delay variation Problem in Ethernet Networks Using SDN Concept", *Advances in Electrical and Electronic Engineering*, vol. 14, no. 4, SI, 397–404, 2016, ISSN: 1336-1376. DOI: 10.15598/aeee.v14i4.1807.

[145] K. Sörensen, M. Sevaux, and F. Glover, "A history of metaheuristics", *CoRR*, vol. abs/1704.00853, 2017. arXiv: 1704.00853. [Online]. Available: http://arxiv.org/abs/1704.00853.

[146] F. Glover and K. Sörensen, *Encyclopedia of Operations Research and Management Science*, S. I. Gass and M. C. Fu, Eds. Springer US, 2013, ISBN: 978-1-4419-1153-7. DOI: 10.1007/978-1-4419-1153-7. [Online]. Available: https://doi.org/10.1007/978-1-4419-1153-7.

[147] K. Sörensen, "Metaheuristics—the metaphor exposed", *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3–18, DOI: 10.1111/itor.12001. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/itor.12001. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12001.

[148] J. Swan, S. Adriaensen, M. Bishr, E. Burke, J. Clark, P. De Causmaecker, J. Durillo, K. Hammond, E. Hart, C. Johnson, Z. Kocsis, B. Kovitz, K. Krawiec, S. Martin, J. Merelo, L. Minku, E. Ozcan, G. Pappa, E. Pesch, P. Garcia-Sanchez, A. Schaerf, K. Sim, J. Smith, T. Stutzle, V. Stefan, S. Wagner, and X. Yao, "A research agenda for metaheuristic standardization", in *MIC 2015: the XI Metaheuristics International Conference*, Agadir, Morocco, Jun. 2015, pp. 1–3.

[149] R. M. H. J. and P. F. Jeffry, "What is a heuristic?", *Computational Intelligence*, vol. 1, no. 1, pp. 47–58, DOI: 10.1111/j.1467-8640.1985.tb00058.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8640.1985.tb00058.x.

[150] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1996, ISBN: 0-262-13316-4.

[151] A. Haghighat, K. Faez, M. Dehghan, A. Mowlaei, and Y. Ghahremani, "Ga-based heuristic algorithms for bandwidth-delay-constrained least-cost multicast routing", *Computer Communications*, vol. 27, no. 1, pp. 111 –127, 2004, ISSN: 0140-3664. DOI: http://dx.doi.org/10.1016/S0140-3664(03)00185-3. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366403001853.

[152] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975, second edition, 1992.

[153] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems.", AAI7609381, PhD thesis, Ann Arbor, MI, USA, 1975.

[154] T. Friedrich, P. S. Oliveto, D. Sudholt, and C. Witt, "Analysis of diversity-preserving mechanisms for global exploration*", *Evol. Comput.*, vol. 17, no. 4, pp. 455–476, Dec. 2009, ISSN: 1063-6560. DOI: 10.1162/evco.2009.17.4.17401. [Online]. Available: http://dx.doi.org/10.1162/evco.2009.17.4.17401.

[155] C. Ravikumar and R. Bajpai, "Source-based delay-bounded multicasting in multimedia networks", *Computer Communications*, vol. 21, no. 2, pp. 126 –132, 1998, ISSN: 0140-3664. DOI: https://doi.org/10.1016/S0140-3664(97)00124-2. [Online]. Available: https://bit.ly/2Z9GFiE.

[156] K. E. Kinnear Jr., Ed., *Advances in Genetic Programming*. Cambridge, MA, USA: MIT Press, 1994, ISBN: 0-262-11188-8.

[157] L. Alvarez, "Design optimization based on genetic programming", PhD thesis, University of Bradford, Bradford, West Yorkshire, UK, 2000.

[158] T. Bäck, "Optimal mutation rates in genetic search", in *Proceedings of the 5th International Conference on Genetic Algorithms*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, pp. 2–8, ISBN: 1-55860-299-2. [Online]. Available: http://dl.acm.org/citation.cfm?id=645513.657408.

[159] S. Meyer-Nieberg and H.-G. Beyer, "Self-adaptation in evolutionary algorithms", in *Parameter Setting in Evolutionary Algorithms*, F. G. Lobo, C. F. Lima, and Z. Michalewicz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 47–75, ISBN: 978-3-540-69432-8. DOI: 10.1007/978-3-540-69432-8_3. [Online]. Available: https://doi.org/10.1007/978-3-540-69432-8_3.

[160] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy", *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, 2012.

[161] T. Storch and I. Wegener, "Real royal road functions for constant population size", in *Proceedings of the 2003 International Conference on Genetic and Evolutionary Computation: PartII*, ser. GECCO'03, Chicago, IL, USA: Springer-Verlag, 2003, pp. 1406–1417, ISBN: 3-540-40603-4. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756582.1756597.

[162] W. M. Spears, "Crossover or mutation?", in *Foundations of Genetic Algorithms*, ser. Foundations of Genetic Algorithms, L. D. WHITLEY, Ed., vol. 2, Elsevier, 1993, pp. 221 –237. DOI: https://doi.org/10.1016/B978-0-08-094832-4.50020-9. [Online]. Available: https://bit.ly/2z25jXL.

[163] N. I. Senaratna, "Genetic algorithms: The crossover-mutation debate", *Bachelor of Computer Science (Special) of the University of Colombo*, 2005. [Online]. Available: https://www-cs.stanford.edu/people/nuwans/docs/GA.pdf.

[164] I. Rechenberg, "Evolutionsstrategie–optimierung technisher systeme nach prinzipien der biologischen evolution", 1973.

[165] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989, ISBN: 0201157675.

[166] C. Arnold, *Evolution runs faster on short timescales*, Q. Magazine, Ed., 2017. [Online]. Available: https://www.quantamagazine.org/evolution-runs-faster-on-short-timescales-20170314.

[167] M. Sipper, W. Fu, K. Ahuja, and J. H. Moore, "Investigating the parameter space of evolutionary algorithms", *BioData Mining*, vol. 11, no. 1, p. 2, 2018, ISSN: 1756-0381. DOI: 10.1186/s13040-018-0164-x. [Online]. Available: https://doi.org/10.1186/s13040-018-0164-x.

[168] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization", in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, ser. NIPS'11, Granada, Spain: Curran Associates Inc., 2011, pp. 2546–2554, ISBN: 978-1-61839-599-3. [Online]. Available: http://dl.acm.org/citation.cfm?id=2986459.2986743.

[169] I. Dewancker, M. McCourt, and S. Clark, *Bayesian optimization primer*, SigOpt, Ed. [Online]. Available: `https://app.sigopt.com/static/pdf/SigOpt_Bayesian_Optimization_Primer.pdf`.

[170] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces", *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, 2001, ISSN: 1573-2916. DOI: `10.1023/A:1012771025575`. [Online]. Available: `https://doi.org/10.1023/A:1012771025575`.

[171] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures", in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ser. ICML'13, Atlanta, GA, USA: JMLR.org, 2013, pp. I–115–I–123. [Online]. Available: `http://dl.acm.org/citation.cfm?id=3042817.3042832`.

[172] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: A python library for model selection and hyperparameter optimization", *Computational Science & Discovery*, vol. 8, no. 1, p. 014008, 2015. DOI: `10.1088/1749-4699/8/1/014008`. [Online]. Available: `https://doi.org/10.1088%2F1749-4699%2F8%2F1%2F014008`.

[173] *Column Generation (GERAD 25TH ANNIVERSARY SERIES)*. Springer, 2005, ISBN: 0387254854. [Online]. Available: `https://www.xarg.org/ref/a/0387254854/`.

[174] M. Kozak, "Efficient control routing and wavelength assignment in loss-less optical burst switching networks", PhD thesis, Faculty of Electrical Engineering, Department of Telecommunication Engineering, Czech Technical University in Prague, 2015. [Online]. Available: `http://hdl.handle.net/10467/61383`.

[175] T. Hegr, J. Vodrazka, and Z. Kocur, "Testing, Troubleshooting and Modelling Tools for Communication Part of Smart Grid", in *2015 Smart Cities Symposium Prague (SCSP)*, Jerabek, M, Ed., Smart Cities Symposium Prague (SCSP), Prague, Czech Republic, Jun 24-25, 2015, IEEE; CTU, Fac Trasportat Sci, 2015, ISBN: 978-1-4673-6727-1.

[176] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment", in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools '08, Marseille, France: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, 60:1–60:10, ISBN: 978-963-9799-20-2. [Online]. Available: `http://dl.acm.org/citation.cfm?id=1416222.1416290`.

[177] INET Core Team, *INET Framework*, 2019. [Online]. Available: `https://inet.omnetpp.org/`.

List of publications

Publications related to the thesis

The percentage is even for all listed authors at each publication.

Impacted journals

- T. Hegr, M. Voznak, M. Kozak, *et al.*, "Measurement of Switching Latency in High Data Rate Ethernet Networks", *Elektronika IR Elektrotechnika*, vol. 21, no. 3, 73–78, 2015, ISSN: 1392-1215. DOI: 10.5755/j01.eee.21.3.10445

  The paper has been cited in:

  - Z. Hu, V. Mukhin, Y. Kornaga, *et al.*, "The Analytical Model for Distributed Computer System Parameters Control Based on Multi-factoring Estimations", *Journal of Network and Systems Management*, vol. 27, no. 2, 351–365, 2019, ISSN: 1064-7570. DOI: 10.1007/s10922-018-9468-x
  - C. Simon, M. Mate, M. Maliosz, *et al.*, "Ethernet with Time Sensitive Networking Tools for Industrial Networks", *Infocommunications Journal*, vol. 9, no. 2, 6–14, 2017, ISSN: 2061-2079

Reviewed journals

- T. Hegr, M. Kozak, and L. Bohac, "On Application of Least-delay variation Problem in Ethernet Networks Using SDN Concept", *Advances in Electrical and Electronic Engineering*, vol. 14, no. 4, SI, 397–404, 2016, ISSN: 1336-1376. DOI: 10.15598/aeee.v14i4.1807

  The paper has been cited in:

  - D. Perepelkin, M. Ivanchikova, V. Byshov, *et al.*, "Development of Architecture of Visual Program System for Distributed Data Processing in Software Defined Networks", in *2018 28th International Conference Radioelektronika (Radioelektronika)*, 28th International Conference on Radioelektronika (RADIOELEKTRONIKA), Prague, CZECH REPUBLIC, APR 19-20, 2018, IEEE Czechoslovakia Sect, 2018, ISBN: 978-1-5386-2485-2

- T. Hegr and L. Bohac, "Impact of nodal centrality measures to robustness in software-defined networking", *Advances in Electrical and Electronic Engineering*, vol. 12, no. 4, pp. 252–259, 2014, ISSN: 1336-1376. DOI: 10.15598/aeee.v12i4.1208

The paper has been cited in:

  - R. Challa, S. Jeon, D. S. Kim, *et al.*, "Centflow: Centrality-based flow balancing and traffic distribution for higher network utilization", *IEEE Access*, vol. 5, pp. 17 045–17 058, 2017

- T. Hegr, L. Bohac, V. Uhlir, *et al.*, "OpenFlow deployment and concept analysis", *Advances in Electrical and Electronic Engineering*, vol. 11, no. 5, pp. 327–335, 2013, ISSN: 1336-1376. DOI: 10.15598/aeee.v11i5.884

  The paper has been cited in:

  - J. Prathima Mabel, K. A. Vani, and K. N. Rama Mohan Babu, "SDN Security: Challenges and Solutions", in *Emerging Research in Electronics, Computer Science and Technology*, V. Sridhar, M. Padma, and K. R. Rao, Eds., Singapore: Springer Singapore, 2019, pp. 837–848, ISBN: 978-981-13-5802-9

Excerpted in WoS

- T. Hegr, J. Vodrazka, and Z. Kocur, "Testing, Troubleshooting and Modelling Tools for Communication Part of Smart Grid", in *2015 Smart Cities Symposium Prague (SCSP)*, Jerabek, M, Ed., Smart Cities Symposium Prague (SCSP), Prague, Czech Republic, Jun 24-25, 2015, IEEE; CTU, Fac Trasportat Sci, 2015, ISBN: 978-1-4673-6727-1

Publications non-related to the thesis

Reviewed journals

- T. Hegr and L. Bohac, "Synthesizing TCP Data Traffic from Industrial Networks for Simulations", *Advances in Electrical and Electronic Engineering*, vol. 13, no. 5, 536–544, 2015, ISSN: 1336-1376. DOI: 10.15598/aeee.v13i5.1501

- M. Klepac, T. Hegr, and L. Bohac, "Enhancing Availability of Services Using Software-Defined Networking", *Advances in Electrical and Electronic Engineering*, vol. 13, no. 5, 529–535, 2015, ISSN: 1336-1376. DOI: 10.15598/aeee.v13i5.1498

Excerpted in WoS

- O. Vondrous, P. Macejko, T. Hegr, *et al.*, "Testing Methodology for Performance Evaluation of Communication Systems for Smart Grid", in *2016 2nd International Conference on Intelligent Green Building and Smart Grid (IGBSG)*, 2nd International Conference on Intelligent Green Building and Smart Grid (IGBSG), Prague, Czech Republic, Jun 27-29, 2016, Czech Tech Univ Prague; Natl Taiwan Univ Sci & Technol; IEEE Czechoslovakia Sect, 2016, 12–17, ISBN: 978-1-4673-8473-5

- V. Hauser and T. Hegr, "Proposal of Adaptive Data Rate Algorithm for LoRaWAN-based Infrastructure", in *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FICLOUD 2017)*, Younas, M and Aleksy, M and Bentahar,

J, Ed., IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), Prague, Czech Republic, AUG 21-23, 2017, IEEE; IEEE Comp Soc Tech Comm Internet; Charles Univ, 2017, 85–90, ISBN: 978-1-5386-2074-8. DOI: 10.1109/FiCloud.2017.47

- R. Kalfus and T. Hegr, "Ultra Narrow Band Radio Technology in High-Density Built-Up Areas", in *Information and Software Technologies, ICIST 2016*, Dregvaite, G and Damasevicius, R, Ed., ser. Communications in Computer and Information Science, 22nd International Conference on Information and Software Technologies (ICIST), Druskininkai, Lithuania, Oct 13-15, 2016, Kaunas Univ Technol, vol. 639, 2016, 663–676, ISBN: 978-3-319-46254-7; 978-3-319-46253-0. DOI: 10.1007/978-3-319-46254-7_54

- O. Vondrous, Z. Kocur, T. Hegr, *et al.*, "Performance Evaluation of IoT Mesh Networking Technology in ISM Frequency Band", in *Proceedings of the 2016 17th International Conference on Mechatronics - Mechatronika (ME) 2016*, Maga, D and Stefek, A and Brezina, T, Ed., 17th International Conference on Mechatronics - Mechatronika (ME), Prague, CZECH REPUBLIC, DEC 07-09, 2016, Czech Tech Univ Prague, Fac Elect Engn; IEEE Czechoslovakia Sect; IEEE Czechoslovakia Sect Ind Applicat Soc Ind Elect Soc Joint Chapter; Brno Univ Technol, Fac Mech Engn; Univ Def Brno, Fac Military Technol, 2016, 488–495, ISBN: 978-8-0010-5883-1

- T. Kukral, M. Kozak, T. Hegr, *et al.*, "VM Migration Measurement and Failure Detection", in *2015 38TH International Conference on Telecommunications and Signal Processing (TSP)*, 38th International Conference on Telecommunications and Signal Processing (TSP), Prague, Czech Republic, Jul 09-11, 2015, IEEE Czechoslovakia Sect; Czech Invest, 2015, 285–288, ISBN: 978-1-4799-8498-5