CONGESTION AVOIDANCE IN INTERNET PROTOCOL (IP) NETWORKS

Dissertation submitted in Partial Fulfillment of the Requirements for Master's Degree in Information and Communication Technologies (ICT)

School of Engineering

University of Zambia

LUGODO NGOMA

513809661

January 2017

DECLARATION

I declare that this dissertation has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where stated otherwise by reference or acknowledgement. The work presented is entirely my own.

.....
Signature

CERTIFICATE OF APPROVAL

This dissertation of Lugodo Ngoma has been approved as fulfilling the requirements for the award of Master of Engineering in Information and Communications Technologies (ICT) by the University of Zambia.

NAME	**SIGNATURE**
Dr. Simon Tembo Supervisor	**…….**
Internal Examiner 1	…….
Internal Examiner 2	……..
Internal Examiner 3	……
Dissertation Chairperson	……

ABSTRACT

As the Internet takes an increasingly central role in our communications infrastructure, the slow convergence of routing protocols after a network failure becomes a growing problem. During failures, the forwarding paths in a network change, the forwarding tables in the routers need to be updated. During the update process, transient link overflows may occur if it is not performed in an orderly manner. Congestion is a problem encountered during transient link updates, therefore to assure congestion avoidance during planned link failures in IP networks, we demonstrate the use of an orderly link cost update algorithm to achieve optimum routing hence avoiding congestion. In this dissertation we present the use of an orderly link cost update method on the Pan European and Germany Network model to avoid congestion during planned link failures. We present a method to obtain the updatable links that do not cause any link overflows. To identify which links need to be orderly updated is a huge permutation problem which ISPs (Internet Service Providers) have been struggling to overcome due to its complexity. Using this method, high load links were established. These are nodes with high link load coupled with high degree order. The objective is to split and move traffic from overloaded links to other links of the network. Using this method, Optimum routing was achieved for the Pan European and Germany network model. The results show that using the proposed link cost update method, there are 6 critical links which can be used to achieve optimum routing. We show that using our approach, we can achieve 100% optimum routing. We use AKITA simulator to implement our algorithm.

Keywords: Congestion, Backup Topology, TCP/IP

ACKNOWLEDGEMENT

My sincere gratitude goes to my Supervisor Dr. Simon Tembo for his patience, guidance and encouragement. This dissertation would not have happened without his support. I would like to go on to thank my fellow graduate students for their invaluable support and suggestions throughout this process. Finally, through this journey I have met many talented and compassionate individuals who did not hesitate to devote their valuable time to me when it was needed. I dare not risk missing to mention anyone's names, so I will simply say "Thank you ALL for being there for me".

TABLE OF CONTENTS

ABSTRACT ..... **i**

ACKNOWLEDGEMENT ..... **ii**

TABLE OF CONTENTS iii

LIST OF TABLES v

LIST OF FIGURES vi

LIST OF ABREVIATIONS viii

CHAPTER 1 - INTRODUCTION **1**

1.1. INTRODUCTION 1

1.2. PROBLEM STATEMENT 2

1.3. OBJECTIVES OF THE RESEARCH 3

1.4. CONCLUSION AND DISSERTATION STRUCTURE 4

CHAPTER 2 – LITERATURE REVIEW **5**

2.1. INTRODUCTION 5

2.2. TCP/IP OVERVIEW 7

2.3. INTERIOR GATEWAY PROTOCOLS (IGP) 10

2.4. LINK STATE ROUTING PROTOCOLS 11

2.5. OSPF OVERVIEW 13

2.6. RELATED WORKS 15

2.7. CONCLUSION 17

CHAPTER 3 – SIMULATION METHODOLOGY **18**

3.1. INTRODUCTION 18

3.2. THOERY 18

3.3. SIMULATION MATERIALS AND ENVIRONMENT 19

3.4. SIMULATION PROCESS 28

CHAPTER 4: RESULTS AND DISCUSSIONS **32**

4.1. INTRODUCTION 32

4.2. 1st LINK COST UPDATE 32

4.3. RESULTS 32

4.3.1. PAN EUROPEAN NETWORK 32

4.3.2. GERMANY NETWORK 37

4.4. DISCUSSION 42

CHAPTER 5:CONCLUSION AND FUTURE WORK 43
5.1. INTRODUCTION 43
5.2. SUMMARY 42
5.3. RECOMMENDATIONS AND FUTURE WORK 45
REFERENCES 46

LIST OF TABLES

3.3.1 Example of permutation results for given updatable links……. 19

3.3.2 Topological parameters…….. 21

3.3.3 Traffic Parameters……. .22

3.3.4 Germany Network topology matrix …….…. .22

3.3.5 Germany voice traffic matrix ……23

3.3.6 Germany data traffic matrix ……...23

3.3.7 Germany IP traffic matrix ……..24

3.3.8 Germany Total traffic matrix ……..……....24

3.3.9 European network traffic matrix ……..….. 25

3.3.10 European voice traffic matrix ……...…25

3.3.11 European data traffic matrix …….....26

3.3.12 European IP traffic matrix ……..…..26

3.3.13 European total traffic matrix …….....27

4.3.1 Permutation results for given updatable Links ……34

LIST OF FIGURES

1.1 Model Showing initial and Target Link costs ……2

1.2 Example showing use of link cost update to avoid link overflow ……....3

2.1 7 layer OSI model ……....5

2.2 Network layer in the reference model …….....6

2.3 TCP/IP protocol stack ……..…..8

2.4 IP Header Format ……..9

2.5.1 Traffic Splitting Ratios in OSPF ……..14

2.5.2 Load Balancing – Proposed Algorithm……....14

3.3.1 Germany network Topology…….20

3.3.2 Pan European network Topology ……21

3.3.3 Updatable link leaf Node Algorithm ……29

3.3.4 An example of a forwarding tree …….…….30

4.3.1 Congestion after 1st LCU ……32

4.3.2 100% optimum routing for the Pan European network ……32

4.3.3 Data Traffic isolation results _ Pan European network ……....33

4.3.4 IP Traffic isolation results _ Pan European network……33

4.3.5 Voice Traffic isolation results _ Pan European network ……..34

4.3.6 Maximum Link loads – Pan European …..……..35

4.3.7Total traffic link utilization _ Pan European network ……..35

4.3.8 100% congestion _ Germany network ……36

4.3.9. No congestion but not optimum _ Germany network ……37

4.3.10. Optimum routing _ Germany network …….……37

4.3.11 Data Traffic Isolation _ Germany ……..…. 38

4.3.12 IP Traffic Isolation _ Germany ……..…. 39

4.3.13 Voice Traffic Isolation _ Germany ……..…… 40

4.3.14 Link Utilization for total traffic Germany ……... 40

LIST OF ABREVIATIONS

BFD – Bidirectional Forwarding detection

CIDR – Classless Inter-Domain Routing

CPU – Central Processing Unit

DNS – Domain Name Server

EGP – Exterior Gateway Protocol

EIGRP – Enhanced Interior Gateway Routing Protocol

FIB – Forwarding Information Base

FTP – File Transfer Protocol

IGP – Interior Gateway Protocol

IGRP - Interior Gateway Routing Protocol

IS – IS – Intermediate system intermediate system

LAN – Local Area Network

LCU - Link Cost Update

OSPF – Open Shortest Path First

RAM - Random Access Memory

RIP – Routing Information Protocol

SDN – Software Defined Networking

SPF – Shortest Path First

OSI – Open system interconnect

Paper

1. Ngoma, L. and Tembo, S., 2015, October. Congestion Avoidance Method in IP Networks. Journal of Natural and Applied Sciences, UNZA.

CHAPTER 1 - INTRODUCTION

1.1. INTRODUCTION

Ideally unlimited bandwidth for every user connected to the network is desired. This is the dream not only of the users but the network engineers also. Reality is however different as we have limitations on the available bandwidth. Limitations that are not only because of the transport protocol but also the medium for transferring information (i.e. Copper wires, Air, etc.). Since resources are limited and user demands are growing exponentially, there is need to control the way users' network resources are used. In today's telecommunications, traffic is converged i.e. data, voice and IP traffic combined. This makes resource sharing in a network a very critical problem in today's networks. Due to stringent Service Level Agreements (SLA) between Service Providers and their customers (Verma, 2004), it is imperative that customers have service at all times. However Service providers need to carry out Routine maintenance from time to time to upgrade, repair or replace some components of their network (Teixeira and Rexford, 2006). During these maintenance works services may be lost for long periods of time. In this research we shall study a congestion avoidance mechanism for planned failures.

Congestion is said to occur in the network when resource demands exceed the capacity and packets are lost due to too much queuing in the network (Jain et al., 1998). During congestion, network throughput may drop to zero and the path delay may become very high. A congestion control scheme helps the network to recover from the congestion state. The problem of congestion control is more difficult to handle in networks with connectionless protocols than in those with connection-oriented protocols.

A congestion avoidance scheme allows a network to operate in the region of low delay and high throughput (Jain et al., 1998). . Such schemes prevent a network from entering the congested state in which packets are lost. Congestion avoidance is a prevention mechanism while congestion control is a recovery mechanism.

1.2. PROBLEM STATEMENT

Performing link cost updates is a permutation problem, but selecting a congestion free link cost update during a transient period for huge permutations is a big challenge for ISPs. The network model shown in figure 1.1 showing the initial and target link costs for traffic engineering purposes is used to illustrate the problem. It can be seen that only 3 links need their link costs updated from 1 to 2, namely links 3 – 7(A), 4 – 6(B) and 1 – 5(C).Therefore there are 3 different links i.e. A, B and C to use in the permutations. The permutations for 3 different links A,B and C is 6 (i.e. ABC,ACB,BCA,BAC,CAB,CBA),therefore there are 6 possible link cost updates when 3 links are updated from the initial link costs to target link costs. However the permutations results become very huge as the number of updatable links increase and finding a sequence order for link cost updates is not practical. Finding such a sequence order is considered NP Hard (Jin et al., 2014).The Simulator used uses a permutation with COST 266 which requires us to identify and update 6 different updatable links. Traffic for a Pan European Network model is simulated using the link cost update algorithm. A network model with traffic matrices and a topology is available. A method to orderly update link costs to avoid link overflows after each link cost update until reaching the target link costs is used. The key idea is identifying the High Load Link, and then selecting a link to which a link cost update can be effected to induce the splitting of traffic to keep the maximum load to a minimum level or/and within the link capacities (Tembo,2013). The aim is to move traffic from congested links to other links of the network. The question is that which 6 links when updated will give us the required optimum routing?

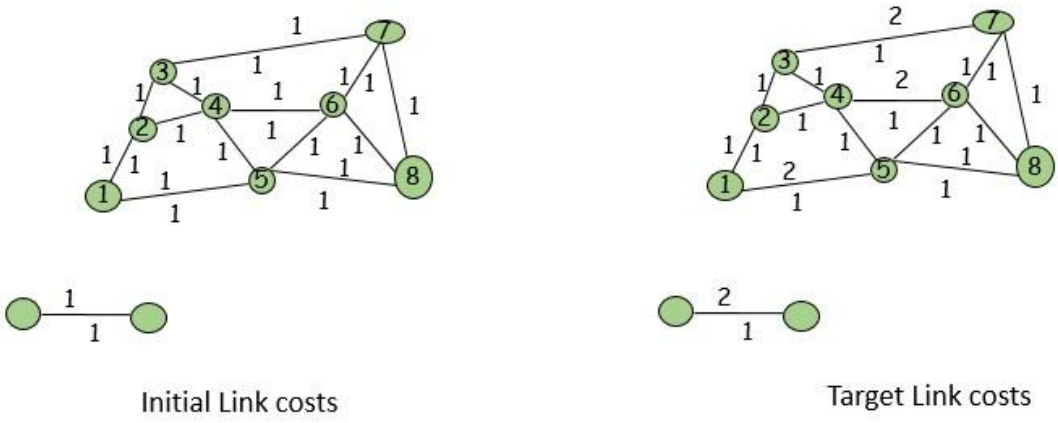

Figure 1.1 Model showing Initial and Target Link costs

For example in figure 1.2 using route optimization method (Fortz and Thorup,2000) whilst depending on the order of link cost updates, the problem of link overflow in link 3 – 7 during the transient state can be avoided. The best solution would be by first updating link 3 – 7 then updating link 1 – 5 to avoid link overflow. However updating link costs of many different links from initial link to target link costs for traffic engineering purposes is a big challenge for ISPs. For Evaluation we use COST 266 (Pan European and Germany network models)

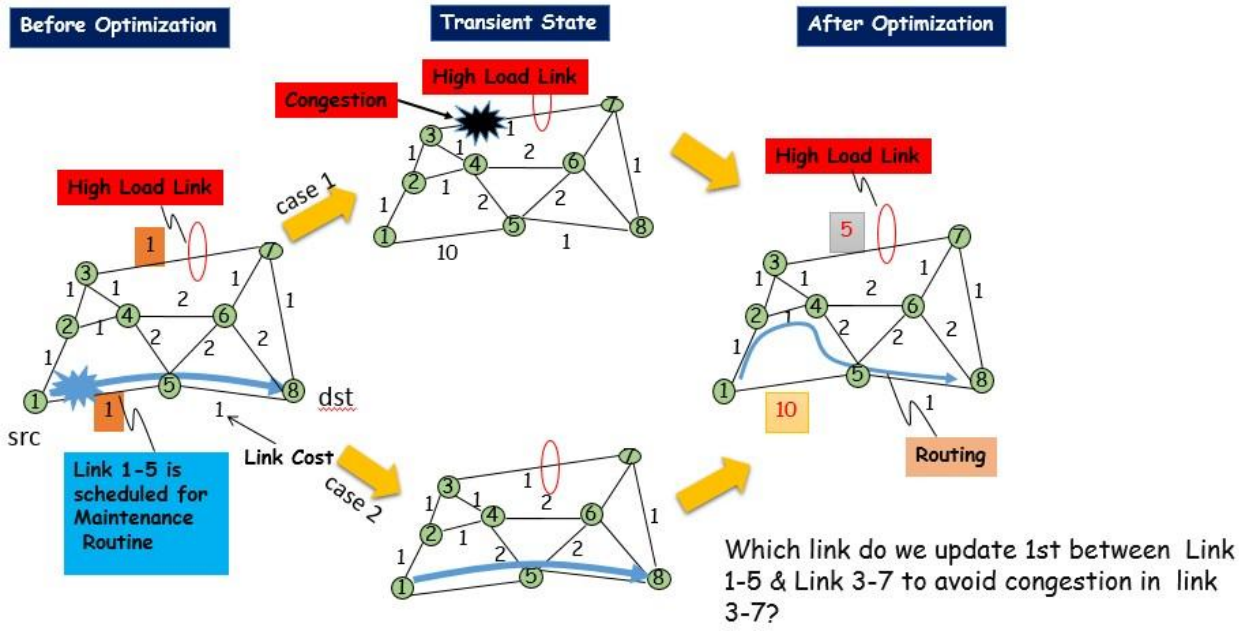

Figure 1.2: Example – Use of Link Cost Updates to avoid Link Overflow (Tembo, 2013)

1.3. OBJECTIVES OF THE RESEARCH

This research intends to show that the proposed algorithm provides a solution to the current problem of congestion avoidance in IP networks in the event of planned failures. Planned failures are type of failures that arise from routine maintenance, network expansion, software upgrades to the systems etc. Furthermore, for most ISPs, fast convergence in case of failures is a key problem that must be solved (Iannaccone et al., 2004), (Francois et al., 2005). Research on congestion avoidance schemes with numerous publications has been conducted by a number of researchers, therefore an analysis of some of these congestion avoidance algorithms. This can be found under the related works section in the simulation methodology section. Lastly for the selected network models namely the Germany and Pan European models, it shall be shown that

using the proposed algorithm, optimum routing can be achieved. The objectives of the research are;

1. Show that optimum routing can be achieved for the network models in question. Note that the proposed method can be used for any network model.
2. Use the updatable link leaf node (ULLN) method to avoid congestion and obtain the links which give us optimum routing i.e. when there is minimum maximum load in the network

The research questions that will be answered in this dissertation are;

1. How does the proposed method avoid congestion?
2. Does the induction of traffic splitting keep the maximum load to a minimum level or within link capacities?
3. Is optimum routing i.e. zero congestion achieved?
4. What is the contribution of this research?

We propose a priority link cost update sequence order method to update link costs to avoid link overflows after each link cost update until reaching the target link costs.

1.4. CONCLUSION AND DISSERTATION STRUCTURE

There have been a lot of studies conducted to investigate the failure recovery after either a link or node failure. This dissertation addresses some of the problems related to failure and congestion recovery. This dissertation provides the analysis and implementation discussion of the algorithm, and presents simulations to test their performance with various types of traffic. Chapter 1 gives an introduction to the problem and gives a subtle distinction between congestion control and congestion avoidance. Chapter 2 looks at the literature review and gives a thorough understanding of the problem .The fundamentals of the interior gateway protocols i.e. OSPF are discussed. Related works are reviewed and I argue how our proposed method stands out among the current congestion avoidance methods. Chapter 3 looks at the simulation methodology used for the research, the algorithm and the simulation process together with the conditions. The results are discussed and analyzed in chapter 4 and the conclusion with future works are presented in the 5^{th} chapter.

CHAPTER 2 – LITERATURE REVIEW

2.1. INTRODUCTION

Configuration changes are a common source of instability in networks, leading to outages, performance disruptions, and security vulnerabilities. Even when the initial and final configurations are correct, the update process itself often steps through intermediate configurations that exhibit incorrect behaviors (Reitblat et al., 2012). Related works in this area will be reviewed. This dissertation looks at interior gateway protocols and how congestion can be avoided during planned failures within a single autonomous system. The basics of interior gateway protocols aswell as an overview of the OSPF routing protocol which is predominately deployed in today's networks will also be reviewed. This dissertation will focus on the network layer which is where congestion occurs.

Figure 2.1. Shows the 7 layers of the OSI model.

Figure 2.1: The 7 layer OSI model (Larry and Bruce, 2003)

A Brief description of the functions of each layer is outlined below;

- Layer 1 (Physical Layer) – The physical layer of the OSI model defines connector and interface specifications, as well as the medium (cable) requirements. Electrical, mechanical, functional, and procedural specifications are provided for sending a bit stream on a computer network.

- Layer 2 (Datalink Layer) – the datalink layer allows a device to access the network to send and receive messages ,offers a physical address so a device's data can be sent on the network ,works with a device's networking software when sending and receiving messages and also Provides error-detection capabilities (Bonaventure,2011).
- Layer 3 (Network Layer) – the network layer of the OSI model, provides an end-to-end logical addressing system so that a packet of data can be routed across several layer 2 networks (Ethernet, Token Ring, Frame Relay, etc.). Note that network layer addresses can also be referred to as logical addresses. These logical addresses are called Internet Protocol addresses (IP). The main focus of our research was on the network layer of the OSI. The main objective of the network layer is to allow end systems, connected to different networks, to exchange information through intermediate systems called routers (Larry and Bruce, 2003). Figure 2.2. Shows the network layer in a reference model. The unit of information in the network layer is called a packet.

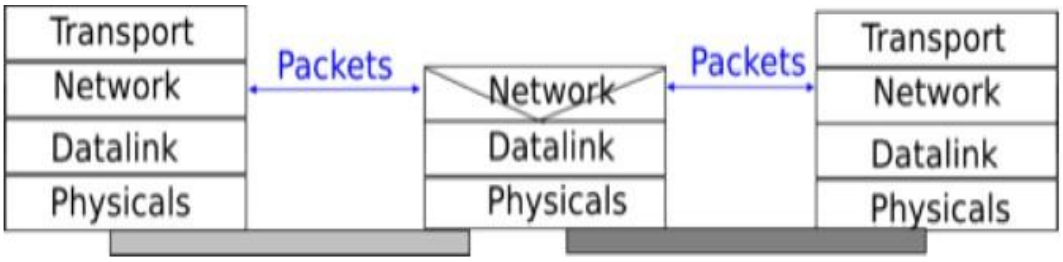

Figure 2.2: The network layer in the reference model (Larry and Bruce, 2003)

- Layer 4 (Transport Layer) – the transport layer of the OSI model, offers end-to-end communication between end devices through a network. Depending on the application, the transport layer either offers reliable, connection-oriented or connectionless, best effort communications (Larry and Bruce, 2003).
- Layer 5 (Session Layer) – the session layer, provides various services, including tracking the number of bytes that each end of the session has acknowledged receiving from the other end of the session (Bonaventure, 2011).

- Layer 6 (Presentation Layer) – the presentation layer is responsible for how an application formats the data to be sent out onto the network. This layer basically allows an application to read (or understand) the message.
- Layer 7 (Application Layer) – the application layer provides an interface for the end user operating a device connected to a network. This layer is what the user sees, in terms of loading an application (such as Web browser or e-mail); that is, this application layer is the data the user views while using these applications.

2.2. TCP/IP OVERVIEW

TCP/IP stands for “Transmission Control Protocol / Internet Protocol” (Hunt,1999). It is basically a network protocol that defines the details of how data is sent and received through network adapters, hubs, switches, routers and other network communications hardware. It was developed by the US department of defense for the purpose of connecting government computer systems to each other through a global, fault tolerant, network (Bonaventure, 2011). The defense department network was opened up to research institutions and eventually the general public to create what is now the Internet. The TCP/IP protocol was also placed in the public domain so that any software company could develop networking software based on the protocol. Because it is the primary protocol used on the Internet, and it is in the public domain, it has become the most popular networking protocol throughout the world and is therefore well supported by almost all computer systems and networking hardware.

The TCP/IP protocol is designed such that each computer or device in a network has a unique "IP Address" (Internet Protocol Address) and each IP address can open and communicate over up to 65535 different “ports” for sending and receiving data to or from any other network device (Clark, 2003). The IP Address uniquely identifies the computer or device on the network and a "Port Number" identifies a specific connection between one computer or device and another (i.e. between two IP Addresses). A TCP/IP "port" can be thought of as a private two-way communications line where the port number is used to identify a unique connection between two devices (Hunt, 1999). The concept is very similar to any other type of port on the PC (serial,

parallel, etc.) except that instead of having a physical connection, the TCP/IP protocol creates a "virtual IP port" and the network hardware and software is responsible for routing data in and out of each virtual IP port. TCP/IP is modeled in layers. This layered representation leads to the term protocol stack, which refers to the stack of layers in the protocol suite. Figure 1.1 shows a layered representation of the TCP/IP protocol stack.

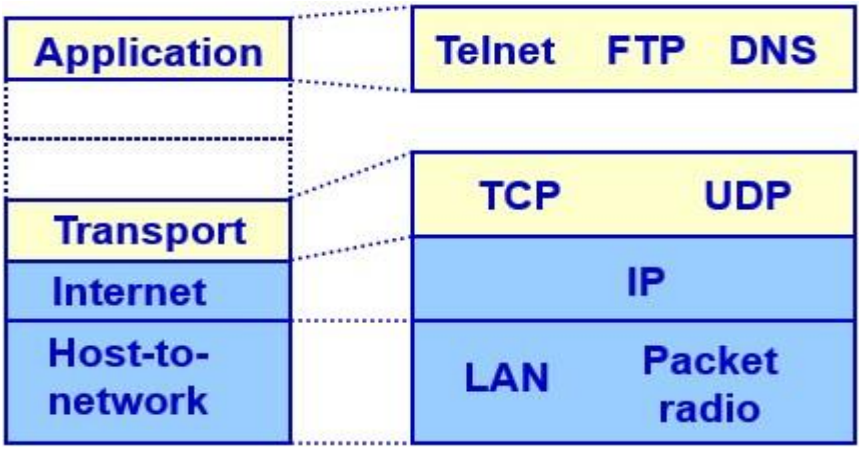

Figure 2.3: The TCP/IP protocol stack: Each layer represents a package of functions (Larry and Bruce, 2003)

TCP is a connection−oriented transport protocol that sends data as an unstructured stream of bytes. By using sequence numbers and acknowledgment messages, TCP can provide a sending node with delivery information about packets transmitted to a destination node. Where data has been lost in transit from source to destination, TCP can retransmit the data until either a timeout condition is reached or until successful delivery has been achieved. TCP can also recognize duplicate messages and will discard them appropriately. If the sending computer is transmitting too fast for the receiving computer, TCP can employ flow control mechanisms to slow data transfer (Gerlach et al., 2004). Internet Protocol (IP) is the most important protocol in the internetwork layer. It is a connectionless protocol that does not assume reliability from lower layers. IP does not provide reliability, flow control, or error recovery. These functions must be provided at a higher level. IP provides a routing function that attempts to deliver transmitted messages to their destination. IP is the protocol that hides the underlying physical network by creating a virtual network view. It is an unreliable, best-effort, and connectionless packet delivery protocol. Note that best-effort means that the packets sent by IP might be lost, arrive

out of order, or even be duplicated. IP assumes higher layer protocols will address these anomalies (Tembo, 2013). The term IP refers to IP version 4. The IPv4 addresses contained in IP packets are 32 bits long. The IP header contains information such as the source or the destination addresses of the packet. There is additional information required at the IP layer besides the source and destination IP addresses. Figure 2.4 below shows the IP header format with the information used at the IP layer to enable reliable delivery of the IP packet.

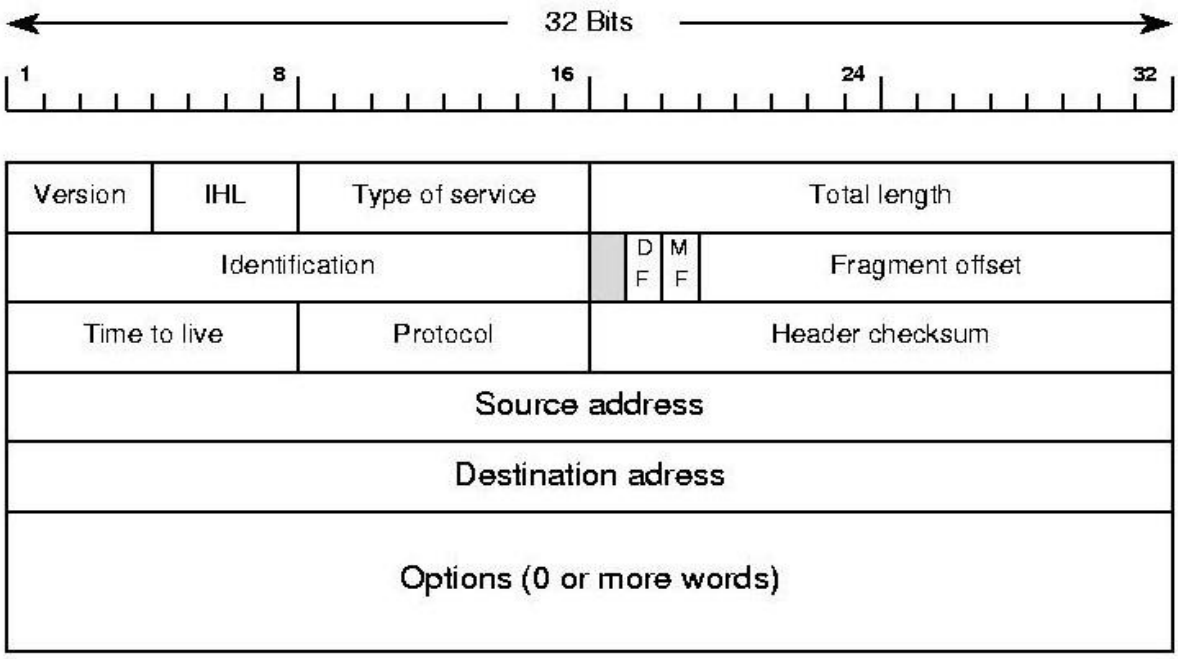

Figure 2.4: IP header Format (Clark, 2003)

Below are some of the descriptions of some fields in the IP header. I shall discuss only the fields relevant to our research.

- **Version**—A 4-bit field that identifies the IP version being used. The current version is 4, and this version is referred to as IPv4.
- **Length**—A 4-bit field containing the length of the IP header in 32-bit increments. The minimum length of an IP header is 20 bytes, or five 32-bit increments. The maximum length of an IP header is 24 bytes, or six 32-bit increments. Therefore, the header length field should contain either 5 or 6 (Clark, 2003)
- **Type of Service (ToS)**—The 8-bit ToS uses 3 bits for IP Precedence, 4 bits for ToS with the last bit not being used. The 4-bit ToS field, although defined, has never been used.

- **Time to Live (TTL)—**It is possible for an IP packet to roam aimlessly around the Internet. If there is a routing problem or a routing loop, then you don't want packets to be forwarded forever. A routing loop is when a packet is continually routed through the same routers over and over (Clark, 2003). The TTL field is initially set to a number and decremented by every router that is passed through. When TTL reaches 0 the packet is discarded.
- **IP Precedence**— A 3-bit field used to identify the level of service a packet receives in the network

2.3. INTERIOR GATEWAY PROTOCOLS (IGP)

Inside each network the intra domain routing protocol, or Interior Gateway Protocol (IGP), is responsible for selecting the path between the ingress router and the egress router. Intra-domain routing protocols compute shortest paths based on "metrics" or "link weights" statically configured by network operators. Network operators usually configure link weights to achieve some traffic-engineering goal such as balancing the traffic load or minimizing propagation delay in the network (Larry and Bruce , 2003). The most common IGPs in large ISPs today are OSPF and IS-IS, which are link-state routing protocols. Each router has a complete view of the network topology. IGP messages are flooded periodically and in response to topology changes, such as link weight changes and equipment going up or down (Clark, 2003). Link weights only change when reconfigured by network operators, not dynamically to adapt to changes in network conditions such as congestion. Each router runs Dijkstra's algorithm to compute the shortest paths in terms of link weights to every other router and uses the results to build the forwarding table. IGPs are used for routing within a routing domain i.e. those networks within the control of a single organization (Bonaventure, 2011). An autonomous system is commonly comprised of many individual networks belonging to companies, schools, and other organizations. An IGP is used to route within the autonomous system, and also used to route within the individual networks themselves (Hunt,1999).

Network operators configure both IGP link weights and BGP policies to control the flow of traffic in the network to achieve good network performance and utilization (Clark, 2003). In this dissertation the focus is on IGPs.

2.4. LINK STATE ROUTING PROTOCOLS

There are two types of routing protocols used predominantly in today's networks namely Distance Vector and Link state routing protocols with the former slowly being phased out (Clark, 2003). Routing protocols like RIP, IGRP, and EIGRP are distance vector routing protocols whereas OSPF and ISIS are classified as link state routing protocols.
All of the routing protocols mentioned above are interior routing protocols (IGP), which means that they are used to exchange routing information within one autonomous system (AS). BGP is an example of an exterior routing protocol (EGP) which is used to exchange routing information between different autonomous systems on the Internet.

As the name implies, distance vector routing protocols use distance to determine the best path to a remote network. The distance is usually the number of hops (routers) to the destination network. Distance vector protocols send complete routing table to each neighbor (a neighbor is a directly connected router that runs the same routing protocol). They usually use some version of Bellman-Ford algorithm to calculate the best routes (Clark, 2003). Distance vector protocols also use more bandwidth because they send complete routing table, while link state protocols sends specific updates only when topology changes occur (Larry and Bruce, 2003). I focus on OSPF routing protocol in this dissertation as it is considered to be simpler due to the fact routing of packets is completely determined by a link cost for each path (Tembo , 2013) .

Link state protocols

These algorithms use the principle of a link state to determine network topology. A link state is the description of an interface on a router (for example, IP address, subnet mask, type of network) and its relationship to neighboring routers (Clark, 2003). The collection of these link states forms a link state database. Link state routing protocols are the second type of routing protocols. They have the same basic purpose as distance vector protocols, to find a best path to a destination, but use different methods to do so. Unlike distance vector protocols, link state protocols don't advertise the entire routing table. Instead, they advertise information about a network topology (directly connected links, neighboring routers...), so that in the end all routers

running a link state protocol have the same topology database. Link state routing protocols converge much faster than distance vector routing protocols, support classless routing, send updates using multicast addresses and use triggered routing updates (Clark , 2003). They also require more router CPU and memory usage than distance-vector routing protocols and can be harder to configure.

Each router running a link state routing protocol creates three different tables:

- Neighbor table - the table of neighboring routers running the same link state routing protocol
- Topology table - the table that stores the topology of the entire network
- Routing table - the table that stores the best routes

Shortest Path First algorithm is used to calculate the best route. OSPF and IS-IS are the link-state (i.e. LS) Interior Gateway Protocols (i.e. IGP) that are used in today's IP networks (Moy,1991). We shall only look at OSPF in this dissertation unless otherwise stated. In link-state routing protocols, each router is configured to;

- Establish and maintain adjacencies with direct neighbors were "Neighbors" are supposed to be able to reach each other at the link-layer.
- Distribute information about the state of these adjacencies through the network i.e. a router constantly checks its ability to reach its configured neighbors using Hello messages (Moy, 1991) or failure detection mechanisms such as BFD - Bidirectional Forwarding Detection (Katz and Ward, 2007).BFD is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols.

This ability is then summarized in a Link-State Packet (IS-IS), or Link-State advertisement (OSPF). This information is then reliably spread through the network, using flooding in IS-IS and OSPF. Routers also distribute in those advertisements reachability information (IP prefixes) assigned to locally connected Local Area Networks (LANs). This information about the state of the adjacencies assist the network operator when there is a failure as they are sent across the network via neighboring nodes. When the link weights are modified, the link state packets are sent to the neighboring nodes. During the update stage, the link state packets are sent to the appropriate neighboring node to inform them of the link weight adjustment.

2.5. OSPF OVERVIEW

The OSPF protocol is based on link-state or SPF (Shortest Path First) technology. This is a departure from the Bellman-Ford base used by traditional TCP/IP internet routing protocols like RIP-Routing Information Protocol, IGRP – Interior Gateway Routing Protocol & EIGRP – Enhanced Interior Gateway Routing Protocol. We use SPF to minimize the maximum load on a link in this dissertation.

OSPF routes IP packets based solely on the destination IP address found in the IP packet header(Moy,1991). IP packets are routed "as is" -- they are not encapsulated in any further protocol headers as they transit the Autonomous System. OSPF is a dynamic routing protocol. It quickly detects topological changes in the AS (such as router interface failures) and calculates new loop-free routes after a period of convergence (Raj and Ibe, 2007). This period of convergence is short and involves a minimum of routing traffic. In a link-state routing protocol, each router maintains a database describing the Autonomous System's topology. This database is referred to as the link-state database. Each participating router has an identical database. Each individual piece of this database is a particular router's local state (e.g., the router's usable interfaces and reachable neighbors). The router distributes its local state throughout the Autonomous System by flooding. OSPF is a link state protocol(Hunt,1999). As with other link state protocols, each OSPF router executes the SPF (shortest Path First) algorithm to process the information stored in the link state database. A better selection of the OSPF link weights can lead to efficient network utilization (Fortz et al.,2002). The algorithm produces a shortest-path tree detailing the preferred routes to each destination network. OSPF provides a number of features not found in distance vector protocols. Support for these features has made OSPF a widely-deployed routing protocol in large networking environments. The following have contributed to the continued acceptance of the OSPF standard:

- Equal cost load balancing: The simultaneous use of multiple paths provides more efficient utilization of network resources. The algorithm used for this research also applies load balancing but in a different way from the traditional OSPF.
- Traditional OSPF equally shares the load without considering network conditions i.e. Utilization levels. The Algorithm used in this research considers congestion before splitting traffic. As shown in figure 2.5.1., Link AC has 60% traffic whilst Link AB was 40%. We have 30% traffic been introduced at node A. Traditionally

OSPF will equally Split the 30% for both Links AB and AC, therefore AC will have 75% percent load whilst AB will have 55%. AC will have a higher load and this can lead to congestion.

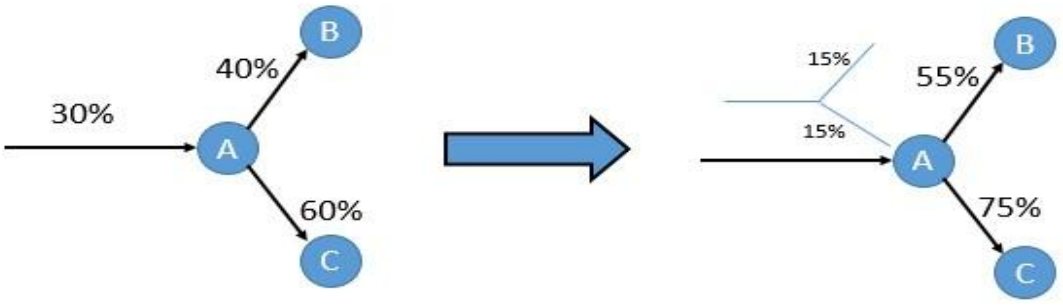

Figure 2.5.1: Traffic splitting ratios under OSPF

The proposed method splits traffic according to link state conditions. Link AC has a 60% traffic load whilst AB. Has 40% traffic as shown in Figure 2.5.2 . The 30% traffic injected in To node A will not be split equally but will be split to attain load balancing as shown in Diagram 2.5.2. It can be seen that the proposed method aims to achieve load balancing after traffic is split and therefore congestion is reduced or avoided. The link costs of the links are used to determine the traffic splitting ratios .During the simulation process, the updating of updatable links was dependent on the link cost of that particular link. The ultimate goal is to achieve minimum maximum load links

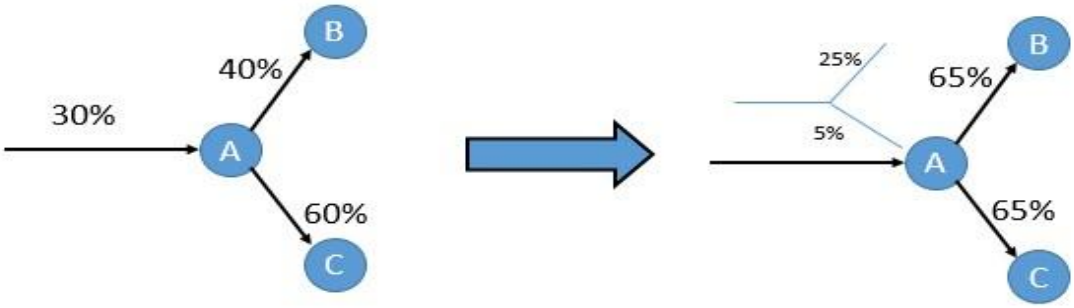

Figure 2.5.2: Load balancing using proposed algorithm

2.6 RELATED WORKS

There have been a number of studies based on different approaches to avoid congestion and link overflows. In (Raza et al., 2009) and (Raza et al., 2011), Raza *et al* propose a theoretical framework to formalize the problem of minimizing a certain disruption function (e.g., link congestion) when the link weights have to be changed. The authors also propose a heuristic to find an ordering in which to modify several IGP weights within a network, so that the number of disruptions is minimal.They look at the sequence of link weight rescheduling whilst we look at link weight splitting. In (Vanbever et al., 2011) Vanbever *et al* proposed a migration strategy that enables operators to perform network-wide changes on an IGP configuration seamlessly, rapidly, and without compromising routing stability. Their strategy relies on effective techniques for the computation of a router migration ordering and on a provisioning system to automate most of the process. In (Reitblat et al., 2011), Reitblat *et al* study the problem of consistent network updates in "Software Defined Networking." They propose a set of consistency properties and show how these properties can be preserved when changes are performed in the network. Unfortunately this work only applies to logically centralized networks. Logically centralized networks refers to SDN – Software defined Networks which is a new paradigm in which a logically-centralized controller manages a collection of programmable switches. The controller responds to events such as topology changes, shifts in traffic load, or new connections from hosts, by pushing forwarding rules to the switches, which process packets efficiently using specialized hardware (Mclurg et al., 2015). Shi *et al* (Shi et al., 2009) propose an update order that avoids transient routing loops and takes the congestion and link overflow into consideration. They study the initial and final forwarding paths to obtain the overflow-free updatable nodes. They also take into consideration a method of analyzing the forwarding tables before and after the updates to establish a loop free order. Francois *et al* (Francois et al., 2007) present a proposal to avoid transient loops by progressively increasing the link cost of the link by one, of times before it can be shut down. This study however concentrates on loop avoidance and not congestion avoidance. More Recent approaches like Zupdate (Liu et al.,2013) and SWAN-Software Driven Wide Area Networks (Hong et al.,2013) attempt to solve the problem by introducing a sequence of intermediate configurations, among which, the update from a former stage to the latter must always be congestion-free. To ensure such a stage sequence exists, they require part of the link capacity to be left vacant, which results in a great waste of link capacities (Zheng et al., 2013). Furthermore, the intermediate configurations they introduce will greatly complicate the update

process, and might even disturb user's QoS (Quality of Service)—e.g., an intermediate path might have a larger latency than both the initial and target ones. Our method in contrast involves identification of a high load link and inducing traffic splitting. This splitting of traffic helps in avoiding capacity wastage. In contrast, Dionysus (Jin et al., 2014) and ATOMIP (Luo et al.,2015) address the challenges by scheduling updates in thoughtful orders without bringing in additional stages in contrast our proposed method induces traffic splitting to keep the maximum load to a minimum level or within link capacities. Dionysus and ATOMIP schedule updates in a thoughtful order without any additional stages. This order arrangement provides a more practical solution, however a congestion free operating sequence may not always exist. The Updates studied in Dionysus and ATOMIP are more concerned with the update order and the update time. Dionysus is a network update scheduler for fast and consistent network updates. Operators have to carefully perform network updates in order to eliminate undesirable behaviors like loops, black holes, policy violations and congestion that may happen during the transient period (Jin et al., 2014). Dionysus is a network update scheduler that adapts to runtime conditions. The intermediate stages introduced using our method – ULLN (Updatable Link Leaf Node) is purely focused on High Load Node identification and Traffic splitting thereby avoiding congestion. This also leads to reduced capacity wastage as all links will be utilized. However despite some similarities in our work with (Raza et al., 2011) and (Hong et al., 2013), our proposed method leads to the identification of 6 links which give us optimum routing at any given moment. The end result is optimization of the network during a planned failure. Also the intermediate stages introduced in these works involve identification of congested links/nodes and selecting the lesser congested links. Our method involves identification of congested links/nodes and inducing traffic splitting to maintain the maximum load at a minimum or within link capacities.

The approaches described have several limitations, as they only work in link-state routing protocols such as OSPF or IS-IS and they only work with single link failures. Our algorithm shows that using appropriate transient link cost update sequence order for link costs for updatable links reduces transient link overflows.

2.7. CONCLUSION

In this chapter, an overview of the routing protocols for conducting the research was given including an overview of link state and distance vectors. A relationship between the layers in the OSI model was also reviewed. However the research main focus is congestion avoidance in IP networks during planned failures. Most research work is centered on congestion control schemes in IP networks. This work is focused on algorithms deployed in networks to reroute traffic during a planned failure. Pre-computation of routes in a network to avoid congestion and routing loops is key**.** The problem is not only to protect against as many possible link failures as possible, but also to use the lowest number of backup FIBs (Forwarding Information Base) as possible. This is different from our study on congestion avoidance during maintenance operations where the changes in network topology and forwarding paths are known. Most of the prior works reviewed focus on the sequencing or ordering of the updates during a failure without necessarily taking congestion avoidance into respect. Those that consider congestion, focus on link weight rescheduling and update time. This leads to capacity wastage for links not being considered for link adjustments as well as increased CPU and memory utilization for the nodes. Link weight rescheduling is similar to link weight splitting but the objective of this research was identification of high load nodes with the subsequent high load links. Link rescheduling involves the sequential identification of link updates. I therefore propose to use the ULLN – updatable link leaf node method for congestion avoidance during a planned failure.

CHAPTER 3: SIMULATION METHODOLOGY

3.1. INTRODUCTION

In this chapter the methods used to conduct the research will be explained. A description of the tools, the materials used and the simulation environment plus the conditions in which the simulations were conducted will be given. The research methods and methodologies will be briefly described. A further highlight on the two main approaches of research classified as qualitative and quantitative approach will be availed, followed by the detailed discussion of the methodology that was used during the research.

The technique used to systematically solve the research problem is known as research methodology. It can also be comprehended as a discipline of studying how research is done logically. Research methods refer to all methods and techniques which are applied during the research conduction. This implies that all those methods which are applied by the researcher in the course of carrying out his research problem are termed as research methods.

3.2. THOERY

In communication networks, a topology is usually a schematic description of the arrangement of a network, including its nodes and connecting lines (Dally and Towles, 2004). There are two ways of defining network geometry: the physical and the logical topologies.

The physical topology of a network is the actual geometric layout of the connecting nodes. There are several common physical topologies, as described below.

Topology can be considered as a virtual shape or structure of a network. The topology does not correspond to the actual arrangement of the devices or actual elements on the network.

Any particular network topology is determined only by the graphical mapping of the configuration of physical and/or logical connections between nodes (Dally and Towles, 2004)

The study and analysis of network topology involve the use of graph theory, distances between nodes, physical interconnections and transmission.

A node in the network has one or more links to one or more nodes in the network. The mapping of these links and nodes in a graph results in a geometric shape that may be used to describe the physical topology of the network. The mapping of the data flow between the nodes in the

network determines the logical topology of the network. The physical and logical topologies may or may not be identical in any particular network.

3.3. SIMULATION MATERIALS AND ENVIRONMENT

The materials used for conducting the simulations were the respective traffic matrices i.e. Voice, IP, Data and Total Traffic obtained from (Gerlach et al., 2004). Total Traffic is an addition of all the 3 types of traffic for each topology. We use 2 different topology models, the Pan European and the Germany network models. We use COST266 (An actual Pan European and Germany network).The models differ in terms of average node degree, path length as illustrated in table 3.3.1. The methodology involves simulation of the Germany and Pan European network topologies using Akita12v2 installed on a Linux Operating system. The Simulations were carried out for Voice, data, IP and total traffic to determine which links carry maximum and minimum traffic and hence determine the most and least congested nodes in the respective topologies. The links with maximum traffic loads will then be selected for updating after which the simulations would be conducted again after updating the maximum load links. The key idea is to induce traffic splitting thereby minimizing the maximum link loads. A forwarding tree diagram would then be constructed using the routing information resulting from each update, this would also help in determining the maximum and minimum link loads. The forwarding tree specifies the forwarding path to a given destination from all routers in the network (Tembo, 2013).Our method provides a Quantitative Evaluation of the Maximum Link Load.

As the number of updatable links increase in a network, the permutations required become astronomical. The table 3.1 gives an example of results for given updatable links

	TOPOLOGY	NODES	LINKS	UPDATABLE LINKS	PERMUTATIONS
1	ABILINE	11	28	3	6
2	COST 239	11	50	6	720
3	COST 239	11	50	14	8.718×10^{10}
4	COST 266	26	100	19	1.216×10^{10}

Table 3.3.1: Example of permutation results for given updatable links (Tembo, 2013)

The simulator used limits the permutations to only 720 thereby giving us only 6 updatable links for optimum routing.

The following materials were used;

A personal computer equipped with the following characteristics was used for the simulations:

- Windows 7 home premium
- System Rating: 2.0 Windows Experience index
- Processor: Intel® Celeron ® CPU 820@ 1.70 GHz
- RAM: 2.00 GB
- System type: 32-bit operating system

The network topologies with respective traffic matrices are shown in the tables and diagrams below;

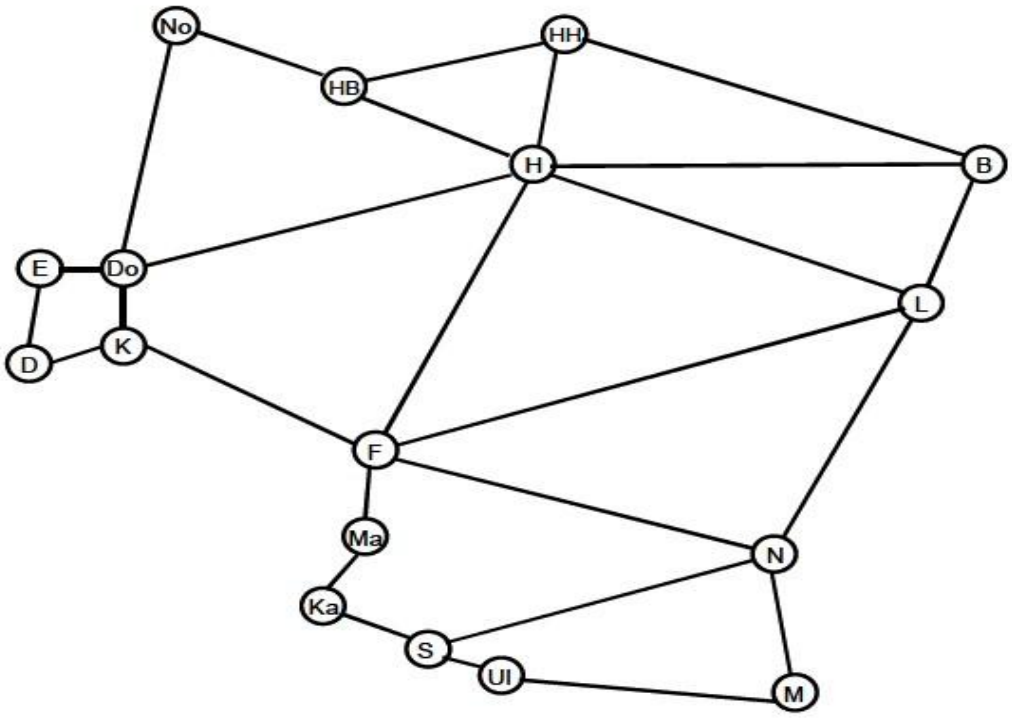

Figure 3.3.1: Germany network (17 nodes) (Gerlach et al., 2004)

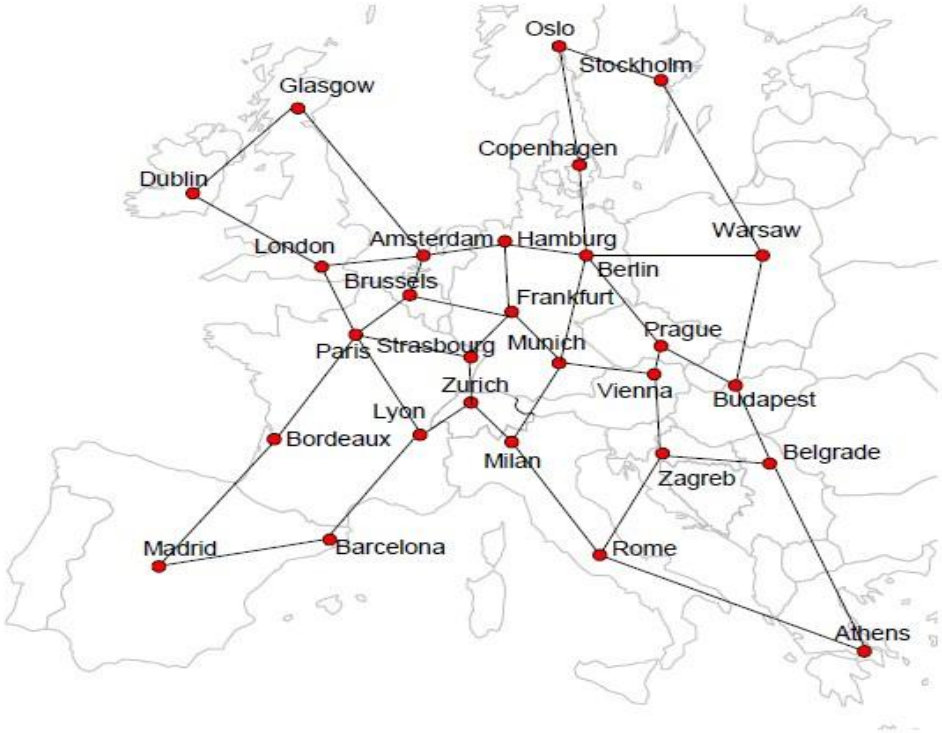

Figure 3.3.2: Pan-European network (28 nodes) (Gerlach et al., 2004)

	German network	European network
number of nodes *n*	17	28
number of links *k*	26	41
minimum node degree	2	2
maximum node degree	6	5
average node degree	3.05882	2.92857
network connectivity	0.191176	0.108466
minimum link length (km)	36	218
maximum link length (km)	353	1500
average link length (km)	170.269	625.366
network diameter (km)	951	5051
average distance (km)	413.5	1983.06
network diameter (hops)	6	8
average distance (hops)	2.69853	3.56085

Table 3.3.2: Topological parameters (Gerlach et al., 2004)

	German network	European network
total traffic volume V	2396.2 Gbit/s	2029.4 Gbit/s
avg. traffic per node	281.9 Gbit/s	145.0 Gbit/s
avg. traffic per node pair	17.6 Gbit/s	5.4 Gbit/s
total traffic load L	6914.2 Gbit/s	6209.6 Gbit/s
average load per link	265.9 Gbit/s	151.5 Gbit/s
average distance per bit	2.89	3.06

Table 3.3.3: Traffic Parameters related to base year 2004 (Gerlach et al., 2004)

Traffic matrices for Germany

The Germany network has 17 nodes with 26 links. The following tables show the various traffic matrices used for simulating Germany network.

GERMANY NETWORK WITH 17 NODES																	
	H	HH	B	L	N	M	UI	S	Ka	Ma	F	K	D	E	Do	No	HB
H		1	1	1	0	0	0	0	0	0	1	0	0	0	1	0	1
HH	1		1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
B	1	1		1	0	0	0	0	0	0	0	0	0	0	0	0	0
L	1	0	1		1	0	0	0	0	0	1	0	0	0	0	0	0
N	0	0	0	1		1	0	1	0	0	1	0	0	0	0	0	0
M	0	0	0	0	1		1	0	0	0	0	0	0	0	0	0	0
UI	0	0	0	0	0	1		1	0	0	0	0	0	0	0	0	0
S	0	0	0	0	1	0	1		1	0	0	0	0	0	0	0	0
Ka	0	0	0	0	0	0	0	1		1	0	0	0	0	0	0	0
Ma	0	0	0	0	0	0	0	0	1		1	0	0	0	0	0	0
F	1	0	0	1	1	0	0	0	0	1		1	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	1		1	0	1	0	0
D	0	0	0	0	0	0	0	0	0	0	0	1		1	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	1		1	0	0
Do	1	0	0	0	0	0	0	0	0	0	0	1	0	1		1	0
No	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		1
HB	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	

Table 3.3.4: Germany network topology matrix

	B	HB	Do	D	E	F	HH	H	Ka	K	L	Ma	M	No	N	S	Ul
B		0.788	0.852	0.828	0.626	2.338	1.974	2.248	0.238	0.886	5.67	0.456	0.884	0	0.932	1.028	0.634
HB	0.788		0.78	0.676	0.556	1.304	2.276	2.394	0.118	0.668	1.132	0.234	0.326	0	0.34	0.468	0.266
Do	0.852	0.78		4.258	5.628	3.34	1.094	1.996	0.27	3.56	1.49	0.57	0.576	0	0.628	0.984	0.516
D	0.828	0.676	4.258		6.408	3.676	1.012	1.654	0.314	8.42	1.44	0.652	0.624	0	0.656	1.108	0.572
E	0.626	0.556	5.628	6.408		2.622	0.798	1.278	0.212	3.61	1.082	0.442	0.442	0	0.47	0.764	0.4
F	2.338	1.304	3.34	3.676	2.622		2.198	3.496	1.53	4.434	4.51	4.426	2.35	9.358	2.898	5.234	2.458
HH	1.974	2.276	1.094	1.012	0.798	2.198		3.73	0.21	1.028	2.436	0.41	0.632	0	0.66	0.854	0.498
H	2.248	2.394	1.996	1.654	1.278	3.496	3.73		0.312	1.664	3.712	0.63	0.874	0	0.988	1.266	0.714
Ka	0.238	0.118	0.27	0.314	0.212	1.53	0.21	0.312		0.384	0.464	0.888	0.38	0	0.376	1.816	0.522
K	0.886	0.668	3.56	8.42	3.61	4.434	1.028	1.664	0.384		1.578	0.82	0.712	0	0.758	1.322	0.67
L	5.67	1.132	1.49	1.44	1.082	4.51	2.436	3.712	0.464	1.578		0.902	1.756	0	2.178	2.044	1.264
Ma	0.456	0.234	0.57	0.652	0.442	4.426	0.41	0.63	0.888	0.82	0.902		0.622	0	0.706	2.122	0.762
M	0.884	0.326	0.576	0.624	0.442	2.35	0.632	0.874	0.38	0.712	1.756	0.622		0	1.798	2.13	2.104
No	0	0	0	0	0	9.358	0	0	0	0	0	0	0		0	0	0
N	0.932	0.34	0.628	0.656	0.47	2.898	0.66	0.988	0.376	0.758	2.178	0.706	1.798	0		2.028	1.402
S	1.028	0.468	0.984	1.108	0.764	5.234	0.854	1.266	1.816	1.322	2.044	2.122	2.13	0	2.028		4.104
Ul	0.634	0.266	0.516	0.572	0.4	2.458	0.498	0.714	0.522	0.67	1.264	0.762	2.104	0	1.402	4.104	

Table 3.3.5: Germany voice traffic matrix (Gbit/s) (Gerlach et al., 2004)

	B	HB	Do	D	E	F	HH	H	Ka	K	L	Ma	M	No	N	S	Ul
B		1.554	1.732	2.036	1.364	6.38	4.086	4.024	0.594	2.166	7.254	1.034	2.386	0	1.882	2.454	1.42
HB	1.554		0.972	1.08	0.754	2.84	2.572	2.436	0.246	1.104	1.902	0.434	0.85	0	0.668	0.972	0.54
Do	1.732	0.972		2.904	2.57	4.708	1.912	2.384	0.398	2.73	2.338	0.728	1.21	0	0.972	1.51	0.806
D	2.036	1.08	2.904		3.27	5.824	2.192	2.588	0.51	5.006	2.74	0.928	1.502	0	1.184	1.91	1.01
E	1.364	0.754	2.57	3.27		3.82	1.498	1.752	0.324	2.524	1.828	0.588	0.974	0	0.772	1.222	0.65
F	6.38	2.84	4.708	5.824	3.82		6.036	6.998	2.024	6.542	8.95	4.194	5.346	27.122	4.544	7.412	3.836
HH	4.086	2.572	1.912	2.192	1.498	6.036		5.05	0.544	2.272	4.63	0.956	1.966	0	1.542	2.18	1.224
H	4.024	2.436	2.384	2.588	1.752	6.998	5.05		0.612	2.668	5.278	1.096	2.136	0	1.742	2.448	1.354
Ka	0.594	0.246	0.398	0.51	0.324	2.024	0.544	0.612		0.582	0.846	0.59	0.64	0	0.488	1.33	0.526
K	2.166	1.104	2.73	5.006	2.524	6.542	2.272	2.668	0.582		2.95	1.07	1.65	0	1.31	2.146	1.124
L	7.254	1.902	2.338	2.74	1.828	8.95	4.63	5.278	0.846	2.95		1.488	3.434	0	2.938	3.532	2.046
Ma	1.034	0.434	0.728	0.928	0.588	4.194	0.956	1.096	0.59	1.07	1.488		1.028	0	0.842	1.812	0.8
M	2.386	0.85	1.21	1.502	0.974	5.346	1.966	2.136	0.64	1.65	3.434	1.028		0	2.224	3.006	2.2
No	0	0	0	0	0	27.122	0	0	0	0	0	0	0		0	0	0
N	1.882	0.668	0.972	1.184	0.772	4.544	1.542	1.742	0.488	1.31	2.938	0.842	2.224	0		2.252	1.38
S	2.454	0.972	1.51	1.91	1.222	7.412	2.18	2.448	1.33	2.146	3.532	1.812	3.006	0	2.252		2.93
Ul	1.42	0.54	0.806	1.01	0.65	3.836	1.224	1.354	0.526	1.124	2.046	0.8	2.2	0	1.38	2.93	

Table 3.3.6: Germany Data traffic matrix (Gbit/s) (Gerlach et al., 2004)

	B	HB	Do	D	E	F	HH	H	Ka	K	L	Ma	M	No	N	S	Ul
B		9.196	13.268	14.618	10.508	49.122	18.778	20.794	4.63	15.652	30.666	8.11	16.46	0	13	19.434	12.152
HB	9.196		5.694	6.274	4.51	21.546	8.06	8.926	1.988	6.718	13.162	3.482	7.066	0	5.58	8.342	5.216
Do	13.268	5.694		9.054	6.508	30.866	11.63	12.878	2.868	9.694	18.99	5.024	10.194	0	8.05	12.036	7.526
D	14.618	6.274	9.054		7.17	33.928	12.814	14.19	3.16	10.68	20.924	5.534	11.232	0	8.87	13.262	8.292
E	10.508	4.51	6.508	7.17		24.564	9.21	10.2	2.27	7.676	15.04	3.978	8.074	0	6.376	9.532	5.96
F	49.122	21.546	30.866	33.928	24.564		43.26	47.73	10.934	36.26	69.144	19.04	38.076	252	30.256	44.718	28.326
HH	18.778	8.06	11.63	12.814	9.21	43.26		18.226	4.058	13.72	26.878	7.11	14.428	0	11.394	17.034	10.652
H	20.794	8.926	12.878	14.19	10.2	47.73	18.226		4.494	15.192	29.764	7.872	15.976	0	12.618	18.864	11.794
Ka	4.63	1.988	2.868	3.16	2.27	10.934	4.058	4.494		3.382	6.626	1.754	3.556	0	2.81	4.2	2.626
K	15.652	6.718	9.694	10.68	7.676	36.26	13.72	15.192	3.382		22.404	5.926	12.026	0	9.498	14.198	8.878
L	30.666	13.162	18.99	20.924	15.04	69.144	26.878	29.764	6.626	22.404		11.608	23.56	0	18.606	27.818	17.394
Ma	8.11	3.482	5.024	5.534	3.978	19.04	7.11	7.872	1.754	5.926	11.608		6.232	0	4.922	7.358	4.6
M	16.46	7.066	10.194	11.232	8.074	38.076	14.428	15.976	3.556	12.026	23.56	6.232		0	9.988	14.932	9.336
No	0	0	0	0	0	252	0	0	0	0	0	0	0		0	0	0
N	13	5.58	8.05	8.87	6.376	30.256	11.394	12.618	2.81	9.498	18.606	4.922	9.988	0		11.792	7.374
S	19.434	8.342	12.036	13.262	9.532	44.718	17.034	18.864	4.2	14.198	27.818	7.358	14.932	0	11.792		11.024
Ul	12.152	5.216	7.526	8.292	5.96	28.326	10.652	11.794	2.626	8.878	17.394	4.6	9.336	0	7.374	11.024	

Table 3.3.7: Germany IP traffic matrix (Gbit/s) (Gerlach et al., 2004)

	B	HB	Do	D	E	F	HH	H	Ka	K	L	Ma	M	No	N	S	Ul
B		11.536	15.85	17.48	12.496	57.834	24.836	27.066	5.46	18.702	43.588	9.6	19.728	0	15.812	22.916	14.204
HB	11.536		7.446	8.028	5.818	25.688	12.906	13.752	2.35	8.488	16.192	4.148	8.24	0	6.586	9.778	6.02
Do	15.85	7.446		16.214	14.704	38.91	14.636	17.258	3.532	15.982	22.816	6.318	11.978	0	9.648	14.528	8.846
D	17.48	8.028	16.214		16.844	43.424	16.014	18.43	3.982	24.104	25.102	7.112	13.354	0	10.708	16.278	9.874
E	12.496	5.818	14.704	16.844		31.004	11.504	13.228	2.804	13.81	17.95	5.006	9.488	0	7.616	11.514	7.008
F	57.834	25.688	38.91	43.424	31.004		51.492	58.224	14.484	47.234	82.6	27.656	45.768	288.48	37.692	57.36	34.614
HH	24.836	12.906	14.636	16.014	11.504	51.492		27.004	4.81	17.018	33.942	8.472	17.022	0	13.594	20.066	12.372
H	27.066	13.752	17.258	18.43	13.228	58.224	27.004		5.414	19.522	38.752	9.596	18.986	0	15.344	22.574	13.86
Ka	5.46	2.35	3.532	3.982	2.804	14.484	4.81	5.414		4.346	7.934	3.228	4.574	0	3.67	7.344	3.672
K	18.702	8.488	15.982	24.104	13.81	47.234	17.018	19.522	4.346		26.928	7.814	14.386	0	11.564	17.666	10.67
L	43.588	16.192	22.816	25.102	17.95	82.6	33.942	38.752	7.934	26.928		13.998	28.748	0	23.722	33.392	20.702
Ma	9.6	4.148	6.318	7.112	5.006	27.656	8.472	9.596	3.228	7.814	13.998		7.88	0	6.468	11.288	6.16
M	19.728	8.24	11.978	13.354	9.488	45.768	17.022	18.986	4.574	14.386	28.748	7.88		0	14.008	20.066	13.638
No	0	0	0	0	0	288.48	0	0	0	0	0	0	0		0	0	0
N	15.812	6.586	9.648	10.708	7.616	37.692	13.594	15.344	3.67	11.564	23.722	6.468	14.008	0		16.07	10.154
S	22.916	9.778	14.528	16.278	11.514	57.36	20.066	22.574	7.344	17.666	33.392	11.288	20.066	0	16.07		18.056
Ul	14.204	6.02	8.846	9.874	7.008	34.614	12.372	13.86	3.672	10.67	20.702	6.16	13.638	0	10.154	18.056	

Table 3.3.8: Germany total traffic matrix (Gbit/s) (Gerlach et al., 2004)

Traffic matrices for Europe

The European network has 28 nodes with 41 links. The following tables show the matrices that were used for simulating the European network.

	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1		1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0		1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1		1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0		1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	1		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0		1	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1		1	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1		1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1		1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1		1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	1		1	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	1		1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		1	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1		0	0	0	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0		1	0	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		1	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1		1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	

Table 3.3.9 European network topology matrix

	Amsterdam	Athens	Barcelona	Belgrade	Berlin	Bordeaux	Brussels	Budapest	Copenhagen	Dublin	Frankfurt	Glasgow	Hamburg	London	Lyon	Madrid	Milan	Munich	Oslo	Paris	Prague	Rome	Stockholm	Strasbourg	Vienna	Warsaw	Zagreb	Zurich
Amsterdam		0.131	0.426	0.200	0.953	0.425	1.573	0.232	0.228	0.136	1.515	1.110	1.493	2.193	0.535	0.357	0.924	0.824	0.130	0.920	0.383	0.589	0.209	0.845	0.232	0.933	0.107	0.315
Athens	0.131		0.187	0.233	0.203	0.121	0.087	0.158	0.044	0.024	0.203	0.183	0.180	0.219	0.147	0.148	0.347	0.244	0.030	0.125	0.118	0.483	0.065	0.150	0.112	0.424	0.072	0.079
Barcelona	0.426	0.187		0.229	0.455	1.093	0.317	0.221	0.100	0.087	0.624	0.585	0.462	0.859	0.921	1.294	1.308	0.647	0.069	0.586	0.249	1.103	0.128	0.535	0.199	0.679	0.119	0.286
Belgrade	0.200	0.233	0.229		0.365	0.158	0.132	0.561	0.071	0.032	0.343	0.247	0.297	0.310	0.213	0.173	0.569	0.471	0.044	0.181	0.244	0.700	0.096	0.248	0.293	0.820	0.210	0.133
Berlin	0.953	0.203	0.455	0.365		0.382	0.540	0.500	0.512	0.101	1.671	0.848	2.796	1.092	0.520	0.365	1.170	1.406	0.183	0.579	1.255	0.832	0.374	0.860	0.536	2.549	0.196	0.373
Bordeaux	0.425	0.121	1.093	0.158	0.382		0.331	0.161	0.087	0.093	0.561	0.584	0.412	0.987	0.840	0.879	0.922	0.510	0.060	0.731	0.199	0.639	0.107	0.480	0.149	0.530	0.083	0.236
Brussels	1.573	0.087	0.317	0.132	0.540	0.331		0.151	0.119	0.085	1.111	0.634	0.717	1.561	0.443	0.257	0.702	0.584	0.070	0.960	0.242	0.416	0.117	0.718	0.152	0.562	0.073	0.251
Budapest	0.232	0.158	0.221	0.561	0.500	0.161	0.151		0.088	0.034	0.423	0.268	0.371	0.340	0.223	0.167	0.605	0.609	0.050	0.198	0.383	0.589	0.112	0.291	0.624	1.172	0.244	0.153
Copenhagen	0.228	0.044	0.100	0.071	0.512	0.087	0.119	0.088		0.028	0.272	0.252	0.632	0.276	0.107	0.085	0.220	0.218	0.083	0.128	0.143	0.166	0.152	0.154	0.083	0.509	0.035	0.067
Dublin	0.136	0.024	0.087	0.032	0.101	0.093	0.085	0.034	0.028		0.122	0.621	0.124	0.416	0.082	0.088	0.131	0.097	0.023	0.123	0.045	0.098	0.035	0.086	0.031	0.135	0.016	0.038
Frankfurt	1.515	0.203	0.624	0.343	1.671	0.561	1.111	0.423	0.272	0.122		0.951	1.798	1.595	0.902	0.472	1.908	2.335	0.139	1.064	0.854	1.027	0.256	2.811	0.468	1.477	0.207	0.824
Glasgow	1.110	0.183	0.585	0.247	0.848	0.584	0.634	0.268	0.252	0.621	0.951		1.074	2.650	0.566	0.571	0.953	0.741	0.222	0.816	0.361	0.720	0.317	0.637	0.246	1.123	0.120	0.278
Hamburg	1.493	0.180	0.462	0.297	2.796	0.412	0.717	0.371	0.632	0.124	1.798	1.074		1.407	0.538	0.382	1.094	1.159	0.216	0.681	0.716	0.752	0.375	0.885	0.377	1.756	0.156	0.359
London	2.193	0.219	0.859	0.310	1.092	0.987	1.561	0.340	0.276	0.416	1.595	2.650	1.407		0.989	0.777	1.477	1.109	0.191	2.132	0.487	0.986	0.304	1.120	0.325	1.305	0.161	0.460
Lyon	0.535	0.147	0.921	0.213	0.520	0.840	0.443	0.223	0.107	0.082	0.902	0.566	0.538	0.989		0.536	2.087	0.882	0.068	0.919	0.292	0.943	0.124	0.941	0.219	0.684	0.124	0.527
Madrid	0.357	0.148	1.294	0.173	0.365	0.879	0.257	0.167	0.085	0.088	0.472	0.571	0.382	0.777	0.536		0.798	0.460	0.062	0.464	0.190	0.694	0.113	0.381	0.149	0.553	0.085	0.192
Milan	0.924	0.347	1.308	0.569	1.170	0.922	0.702	0.605	0.220	0.131	1.908	0.953	1.094	1.477	2.087	0.798		2.831	0.132	1.104	0.755	2.852	0.255	1.948	0.620	1.597	0.391	1.585
Munich	0.824	0.244	0.647	0.471	1.406	0.510	0.584	0.609	0.218	0.097	2.335	0.741	1.159	1.109	0.882	0.460	2.831		0.117	0.742	1.168	1.411	0.231	1.782	0.786	1.622	0.354	1.036
Oslo	0.130	0.030	0.069	0.044	0.183	0.060	0.070	0.050	0.083	0.023	0.139	0.222	0.216	0.191	0.068	0.062	0.132	0.117		0.082	0.068	0.106	0.158	0.086	0.045	0.268	0.020	0.038
Paris	0.920	0.125	0.586	0.181	0.579	0.731	0.960	0.198	0.128	0.123	1.064	0.816	0.681	2.132	0.919	0.464	1.104	0.742	0.082		0.285	0.637	0.141	0.911	0.194	0.690	0.099	0.364
Prague	0.383	0.118	0.249	0.244	1.255	0.199	0.242	0.383	0.143	0.045	0.854	0.361	0.716	0.487	0.292	0.190	0.755	1.168	0.068	0.285		0.528	0.143	0.493	0.552	1.261	0.153	0.235
Rome	0.589	0.483	1.103	0.700	0.832	0.639	0.416	0.589	0.166	0.098	1.027	0.720	0.752	0.986	0.943	0.694	2.852	1.411	0.106	0.637	0.528		0.213	0.849	0.507	1.388	0.400	0.504
Stockholm	0.209	0.065	0.128	0.096	0.374	0.107	0.117	0.112	0.152	0.035	0.256	0.317	0.375	0.304	0.124	0.113	0.255	0.231	0.158	0.141	0.143	0.213		0.159	0.096	0.697	0.043	0.073
Strasbourg	0.845	0.150	0.535	0.248	0.860	0.480	0.718	0.291	0.154	0.086	2.811	0.637	0.885	1.120	0.941	0.381	1.948	1.782	0.086	0.911	0.493	0.849	0.159		0.315	0.924	0.155	1.225
Vienna	0.232	0.112	0.199	0.293	0.536	0.149	0.152	0.624	0.083	0.031	0.468	0.246	0.377	0.325	0.219	0.149	0.620	0.786	0.045	0.194	0.552	0.507	0.096	0.315		0.933	0.222	0.166
Warsaw	0.933	0.424	0.679	0.820	2.549	0.530	0.562	1.172	0.509	0.135	1.477	1.123	1.756	1.305	0.684	0.553	1.597	1.622	0.268	0.690	1.261	1.388	0.697	0.924	0.933		0.348	0.443
Zagreb	0.107	0.072	0.119	0.210	0.196	0.083	0.073	0.244	0.035	0.016	0.207	0.120	0.156	0.161	0.124	0.085	0.391	0.354	0.020	0.099	0.153	0.400	0.043	0.155	0.222	0.348		0.088
Zurich	0.315	0.079	0.286	0.133	0.373	0.236	0.251	0.153	0.067	0.038	0.824	0.278	0.359	0.460	0.527	0.192	1.585	1.036	0.038	0.364	0.235	0.504	0.073	1.225	0.166	0.443	0.088	
Demand [Gbit/s]																												

Table 3.3.10: Europe voice traffic matrix (Gbit/s) (Gerlach et al., 2004)

	Amsterdam	Athens	Barcelona	Belgrade	Berlin	Bordeaux	Brussels	Budapest	Copenhagen	Dublin	Frankfurt	Glasgow	Hamburg	London	Lyon	Madrid	Milan	Munich	Oslo	Paris	Prague	Rome	Stockholm	Strasbourg	Vienna	Warsaw	Zagreb	Zurich
Amsterdam		1.164	2.874	0.530	5.281	2.622	4.116	1.556	1.425	0.829	6.659	6.835	6.610	9.609	2.940	2.629	5.104	4.911	0.988	3.857	2.460	4.074	1.655	3.697	1.491	6.506	0.779	1.959
Athens	1.164		1.408	0.424	1.801	1.036	0.715	0.949	0.464	0.257	1.803	2.052	1.700	2.248	1.140	1.253	2.313	1.977	0.353	1.051	1.009	2.731	0.683	1.153	0.768	3.244	0.471	0.727
Barcelona	2.874	1.408		0.574	3.688	4.251	1.868	1.533	0.955	0.669	4.322	5.017	3.720	6.083	3.903	5.061	6.141	4.399	0.727	3.112	2.006	5.641	1.311	2.973	1.398	5.612	0.828	1.885
Belgrade	0.530	0.424	0.574		0.891	0.436	0.325	0.659	0.217	0.109	0.864	0.879	0.803	0.984	0.506	0.498	1.092	1.012	0.157	0.466	0.535	1.211	0.307	0.546	0.457	1.663	0.297	0.347
Berlin	5.281	1.801	3.688	0.891		3.090	2.997	2.835	2.651	0.887	8.690	7.425	11.24	8.427	3.604	3.304	7.141	7.973	1.457	3.803	5.532	6.022	2.755	4.635	2.817	13.37	1.308	2.648
Bordeaux	2.622	1.036	4.251	0.436	3.090		1.746	1.195	0.811	0.631	3.744	4.583	3.209	5.957	3.405	3.812	4.712	3.569	0.620	3.177	1.637	3.923	1.094	2.575	1.105	4.533	0.632	1.565
Brussels	4.116	0.715	1.868	0.325	2.997	1.746		0.945	0.774	0.494	4.300	3.895	3.454	6.114	2.018	1.682	3.356	3.118	0.547	2.971	1.475	2.583	0.936	2.569	0.909	3.809	0.483	1.319
Budapest	1.556	0.949	1.533	0.659	2.835	1.195	0.945		0.656	0.307	2.608	2.493	2.444	2.807	1.410	1.335	3.062	3.131	0.456	1.326	1.823	3.023	0.899	1.608	1.814	5.407	0.872	1.011
Copenhagen	1.425	0.464	0.955	0.217	2.651	0.811	0.774	0.656		0.257	1.933	2.234	2.946	2.335	0.901	0.881	1.706	1.729	0.541	0.986	1.029	1.484	0.968	1.081	0.612	3.292	0.303	0.618
Dublin	0.829	0.257	0.669	0.109	0.887	0.631	0.494	0.307	0.257		0.977	2.638	0.982	2.159	0.594	0.675	0.991	0.868	0.213	0.726	0.435	0.858	0.350	0.607	0.282	1.278	0.154	0.350
Frankfurt	6.659	1.803	4.322	0.864	8.690	3.744	4.300	2.608	1.933	0.977		7.863	9.014	10.18	4.748	3.758	9.118	10.27	1.273	5.156	4.564	6.688	2.277	8.379	2.634	10.18	1.346	3.935
Glasgow	6.835	2.052	5.017	0.879	7.425	4.583	3.895	2.493	2.234	2.638	7.863		8.357	15.75	4.510	4.959	7.729	6.942	1.928	5.416	3.558	6.720	3.041	4.784	2.289	10.64	1.229	2.742
Hamburg	6.610	1.700	3.720	0.803	11.24	3.209	3.454	2.444	2.946	0.982	9.014	8.357		9.565	3.665	3.379	6.905	7.238	1.584	4.125	4.179	5.725	2.758	4.702	2.365	11.09	1.169	2.598
London	9.609	2.248	6.083	0.984	8.427	5.957	6.114	2.807	2.335	2.159	10.18	15.75	9.565		5.962	5.783	9.622	8.491	1.786	8.755	4.135	7.862	2.980	6.345	2.633	11.47	1.422	3.527
Lyon	2.940	1.140	3.903	0.506	3.604	3.405	2.018	1.410	0.901	0.594	4.748	4.510	3.665	5.962		2.978	7.089	4.693	0.659	3.562	1.983	4.765	1.180	3.605	1.338	5.146	0.774	2.341
Madrid	2.629	1.253	5.061	0.498	3.304	3.812	1.682	1.335	0.881	0.675	3.758	4.959	3.379	5.783	2.978		4.797	3.708	0.689	2.769	1.753	4.474	1.229	2.509	1.207	5.064	0.700	1.544
Milan	5.104	2.313	6.141	1.092	7.141	4.712	3.356	3.062	1.706	0.991	9.118	7.729	6.905	9.622	7.089	4.797		11.11	1.218	5.155	4.213	10.95	2.234	6.850	2.976	10.39	1.814	5.358
Munich	4.911	1.977	4.399	1.012	7.973	3.569	3.118	3.131	1.729	0.868	10.27	6.942	7.238	8.491	4.693	3.708	11.11		1.166	4.306	5.336	7.840	2.164	6.672	3.412	10.66	1.760	4.412
Oslo	0.988	0.353	0.727	0.157	1.457	0.620	0.547	0.456	0.541	0.213	1.273	1.928	1.584	1.786	0.659	0.689	1.218	1.166		0.725	0.653	1.091	0.908	0.744	0.413	2.199	0.213	0.431
Paris	3.857	1.051	3.112	0.466	3.803	3.177	2.971	1.326	0.986	0.726	5.156	5.416	4.125	8.755	3.562	2.769	5.155	4.306	0.725		1.959	3.916	1.257	3.546	1.260	5.171	0.693	1.945
Prague	2.460	1.009	2.006	0.535	5.532	1.637	1.475	1.823	1.029	0.435	4.564	3.558	4.179	4.135	1.983	1.753	4.213	5.336	0.653	1.959		3.524	1.249	2.577	2.102	6.905	0.848	1.543
Rome	4.074	2.731	5.641	1.211	6.022	3.923	2.583	3.023	1.484	0.858	6.688	6.720	5.725	7.862	4.765	4.474	10.95	7.840	1.091	3.916	3.524		2.042	4.522	2.691	9.684	1.835	3.021
Stockholm	1.655	0.683	1.311	0.307	2.755	1.094	0.936	0.899	0.968	0.350	2.277	3.041	2.758	2.960	1.180	1.229	2.234	2.164	0.908	1.257	1.249	2.042		1.335	0.801	4.681	0.409	0.783
Strasbourg	3.697	1.153	2.973	0.546	4.635	2.575	2.569	1.608	1.081	0.607	8.379	4.784	4.702	6.345	3.605	2.509	6.850	6.672	0.744	3.546	2.577	4.522	1.335		1.607	5.984	0.865	3.567
Vienna	1.491	0.768	1.398	0.457	2.817	1.105	0.909	1.814	0.612	0.282	2.634	2.289	2.365	2.633	1.338	1.207	2.976	3.412	0.413	1.260	2.102	2.691	0.801	1.607		4.631	0.798	1.012
Warsaw	6.506	3.244	5.612	1.663	13.37	4.533	3.809	5.407	3.292	1.278	10.18	10.64	11.09	11.47	5.146	5.064	10.39	10.66	2.199	5.171	6.905	9.684	4.681	5.984	4.631		2.172	3.595
Zagreb	0.779	0.471	0.828	0.297	1.308	0.632	0.483	0.872	0.303	0.154	1.346	1.229	1.169	1.422	0.774	0.700	1.814	1.760	0.213	0.693	0.848	1.835	0.409	0.865	0.798	2.172		0.567
Zurich	1.959	0.727	1.885	0.347	2.648	1.565	1.319	1.011	0.618	0.350	3.935	2.742	2.598	3.527	2.341	1.544	5.358	4.412	0.431	1.945	1.543	3.021	0.783	3.567	1.012	3.595	0.567	

Table 3.3.11: Europe Data Traffic matrix (Gbit/s) (Gerlach et al., 2004)

	Amsterdam	Athens	Barcelona	Belgrade	Berlin	Bordeaux	Brussels	Budapest	Copenhagen	Dublin	Frankfurt	Glasgow	Hamburg	London	Lyon	Madrid	Milan	Munich	Oslo	Paris	Prague	Rome	Stockholm	Strasbourg	Vienna	Warsaw	Zagreb	Zurich
Amsterdam		0.851	2.292	0.327	4.256	1.833	1.768	0.458	1.048	0.655	4.256	10.80	4.256	10.80	1.833	2.292	3.601	4.256	1.440	1.833	1.440	3.601	2.946	1.833	1.768	3.208	0.196	2.226
Athens	0.851		0.438	0.063	0.814	0.350	0.338	0.088	0.200	0.125	0.814	2.065	0.814	2.065	0.350	0.438	0.688	0.814	0.275	0.350	0.275	0.688	0.563	0.350	0.338	0.613	0.038	0.426
Barcelona	2.292	0.438		0.168	2.190	0.944	0.910	0.236	0.539	0.337	2.190	5.560	2.190	5.560	0.944	1.179	1.853	2.190	0.741	0.944	0.741	1.853	1.516	0.944	0.910	1.651	0.101	1.146
Belgrade	0.327	0.063	0.168		0.313	0.135	0.130	0.034	0.077	0.048	0.313	0.794	0.313	0.794	0.135	0.168	0.265	0.313	0.106	0.135	0.106	0.265	0.217	0.135	0.130	0.236	0.014	0.164
Berlin	4.256	0.814	2.190	0.313		1.752	1.690	0.438	1.001	0.626	4.068	10.33	4.07	10.33	1.752	2.190	3.442	4.068	1.377	1.752	1.377	3.442	2.816	1.752	1.690	3.07	0.188	2.128
Bordeaux	1.833	0.350	0.944	0.135	1.752		0.728	0.189	0.431	0.270	1.752	4.448	1.752	4.448	0.755	0.944	1.483	1.752	0.593	0.755	0.593	1.483	1.213	0.755	0.728	1.321	0.081	0.917
Brussels	1.768	0.338	0.910	0.130	1.690	0.728		0.182	0.416	0.260	1.690	4.289	1.690	4.289	0.728	0.910	1.430	1.690	0.572	0.728	0.572	1.430	1.170	0.728	0.702	1.274	0.078	0.884
Budapest	0.458	0.088	0.236	0.034	0.438	0.189	0.182		0.108	0.067	0.438	1.112	0.438	1.112	0.189	0.236	0.371	0.438	0.148	0.189	0.148	0.371	0.303	0.189	0.182	0.330	0.020	0.229
Copenhagen	1.048	0.200	0.539	0.077	1.001	0.431	0.416	0.108		0.154	1.001	2.542	1.001	2.542	0.431	0.539	0.847	1.001	0.339	0.431	0.339	0.847	0.693	0.431	0.416	0.755	0.046	0.524
Dublin	0.655	0.125	0.337	0.048	0.626	0.270	0.260	0.067	0.154		0.626	1.589	0.626	1.589	0.270	0.337	0.530	0.626	0.212	0.270	0.212	0.530	0.433	0.270	0.260	0.472	0.029	0.327
Frankfurt	4.256	0.814	2.190	0.313	4.068	1.752	1.690	0.438	1.001	0.626		10.33	4.068	10.33	1.752	2.190	3.442	4.07	1.377	1.752	1.377	3.442	2.816	1.752	1.690	3.07	0.188	2.128
Glasgow	10.80	2.065	5.560	0.794	10.33	4.448	4.289	1.112	2.542	1.589	10.33		10.33	26.21	4.448	5.560	8.738	10.33	3.495	4.448	3.495	8.738	7.149	4.448	4.289	7.78	0.477	5.401
Hamburg	4.256	0.814	2.190	0.313	4.07	1.752	1.690	0.438	1.001	0.626	4.068	10.33		10.33	1.752	2.190	3.442	4.068	1.377	1.752	1.377	3.442	2.816	1.752	1.690	3.07	0.188	2.128
London	10.80	2.065	5.560	0.794	10.33	4.448	4.289	1.112	2.542	1.589	10.33	26.21	10.33		4.448	5.560	8.738	10.33	3.495	4.448	3.495	8.738	7.149	4.448	4.289	7.78	0.477	5.401
Lyon	1.833	0.350	0.944	0.135	1.752	0.755	0.728	0.189	0.431	0.270	1.752	4.448	1.752	4.448		0.944	1.483	1.752	0.593	0.755	0.593	1.483	1.213	0.755	0.728	1.321	0.081	0.917
Madrid	2.292	0.438	1.179	0.168	2.190	0.944	0.910	0.236	0.539	0.337	2.190	5.560	2.190	5.560	0.944		1.853	2.190	0.741	0.944	0.741	1.853	1.516	0.944	0.910	1.651	0.101	1.146
Milan	3.601	0.688	1.853	0.265	3.442	1.483	1.430	0.371	0.847	0.530	3.442	8.738	3.442	8.738	1.483	1.853		3.44	1.165	1.483	1.165	2.91	2.383	1.483	1.430	2.59	0.159	1.800
Munich	4.256	0.814	2.190	0.313	4.068	1.752	1.690	0.438	1.001	0.626	4.07	10.33	4.068	10.33	1.752	2.190	3.44		1.377	1.752	1.377	3.442	2.816	1.752	1.690	3.07	0.188	2.128
Oslo	1.440	0.275	0.741	0.106	1.377	0.593	0.572	0.148	0.339	0.212	1.377	3.495	1.377	3.495	0.593	0.741	1.165	1.377		0.593	0.466	1.165	0.953	0.593	0.572	1.038	0.064	0.720
Paris	1.833	0.350	0.944	0.135	1.752	0.755	0.728	0.189	0.431	0.270	1.752	4.448	1.752	4.448	0.755	0.944	1.483	1.752	0.593		0.593	1.483	1.213	0.755	0.728	1.321	0.081	0.917
Prague	1.440	0.275	0.741	0.106	1.377	0.593	0.572	0.148	0.339	0.212	1.377	3.495	1.377	3.495	0.593	0.741	1.165	1.377	0.466	0.593		1.165	0.953	0.593	0.572	1.038	0.064	0.720
Rome	3.601	0.688	1.853	0.265	3.442	1.483	1.430	0.371	0.847	0.530	3.442	8.738	3.442	8.738	1.483	1.853	2.91	3.442	1.165	1.483	1.165		2.383	1.483	1.430	2.595	0.159	1.800
Stockholm	2.946	0.563	1.516	0.217	2.816	1.213	1.170	0.303	0.693	0.433	2.816	7.149	2.816	7.149	1.213	1.516	2.383	2.816	0.953	1.213	0.953	2.383		1.213	1.170	2.123	0.130	1.473
Strasbourg	1.833	0.350	0.944	0.135	1.752	0.755	0.728	0.189	0.431	0.270	1.752	4.448	1.752	4.448	0.755	0.944	1.483	1.752	0.593	0.755	0.593	1.483	1.213		0.728	1.321	0.081	0.917
Vienna	1.768	0.338	0.910	0.130	1.690	0.728	0.702	0.182	0.416	0.260	1.690	4.289	1.690	4.289	0.728	0.910	1.430	1.690	0.572	0.728	0.572	1.430	1.170	0.728		1.274	0.078	0.884
Warsaw	3.208	0.613	1.651	0.236	3.07	1.321	1.274	0.330	0.755	0.472	3.07	7.78	3.07	7.78	1.321	1.651	2.59	3.07	1.038	1.321	1.038	2.595	2.123	1.321	1.274		0.142	1.604
Zagreb	0.196	0.038	0.101	0.014	0.188	0.081	0.078	0.020	0.046	0.029	0.188	0.477	0.188	0.477	0.081	0.101	0.159	0.188	0.064	0.081	0.064	0.159	0.130	0.081	0.078	0.142		0.098
Zurich	2.226	0.426	1.146	0.164	2.128	0.917	0.884	0.229	0.524	0.327	2.128	5.401	2.128	5.401	0.917	1.146	1.800	2.128	0.720	0.917	0.720	1.800	1.473	0.917	0.884	1.604	0.098	

Table 3.3.12: Europe IP Traffic matrix (Gbit/s) (Gerlach et al., 2004)

	Amsterdam	Athens	Barcelona	Belgrade	Berlin	Bordeaux	Brussels	Budapest	Copenhagen	Dublin	Frankfurt	Glasgow	Hamburg	London	Lyon	Madrid	Milan	Munich	Oslo	Paris	Prague	Rome	Stockholm	Strasbourg	Vienna	Warsaw	Zagreb	Zurich
Amsterdam		2.146	5.591	1.057	10.49	4.881	7.457	2.247	2.701	1.620	12.43	18.75	12.36	22.60	5.308	5.277	9.628	9.991	2.558	6.611	4.283	8.264	4.810	6.376	3.491	10.65	1.082	4.501
Athens	2.146		2.033	0.719	2.818	1.508	1.139	1.195	0.709	0.407	2.819	4.300	2.694	4.533	1.638	1.839	3.349	3.035	0.659	1.526	1.403	3.903	1.311	1.653	1.218	4.281	0.580	1.231
Barcelona	5.591	2.033		0.971	6.333	6.288	3.095	1.990	1.595	1.093	7.136	11.16	6.372	12.50	5.768	7.535	9.302	7.236	1.538	4.642	2.997	8.598	2.955	4.451	2.507	7.942	1.048	3.316
Belgrade	1.057	0.719	0.971		1.569	0.728	0.586	1.253	0.365	0.189	1.520	1.921	1.413	2.088	0.854	0.839	1.926	1.795	0.306	0.781	0.885	2.175	0.620	0.928	0.880	2.720	0.522	0.644
Berlin	10.49	2.818	6.333	1.569		5.225	5.226	3.773	4.164	1.614	14.43	18.60	18.11	19.84	5.877	5.859	11.75	13.45	3.017	6.135	8.163	10.30	5.946	7.247	5.043	18.99	1.691	5.148
Bordeaux	4.881	1.508	6.288	0.728	5.225		2.805	1.544	1.330	0.994	6.057	9.615	5.373	11.39	4.999	5.634	7.117	5.831	1.273	4.663	2.429	6.044	2.414	3.810	1.981	6.384	0.796	2.717
Brussels	7.457	1.139	3.095	0.586	5.226	2.805		1.278	1.309	0.839	7.101	8.818	5.860	11.96	3.189	2.848	5.488	5.392	1.189	4.659	2.289	4.428	2.223	4.015	1.763	5.645	0.634	2.454
Budapest	2.247	1.195	1.990	1.253	3.773	1.544	1.278		0.851	0.409	3.468	3.874	3.253	4.260	1.822	1.739	4.038	4.178	0.654	1.712	2.354	3.983	1.314	2.087	2.619	6.909	1.136	1.393
Copenhagen	2.701	0.709	1.595	0.365	4.164	1.330	1.309	0.851		0.439	3.207	5.028	4.580	5.153	1.439	1.505	2.773	2.948	0.963	1.546	1.510	2.498	1.814	1.666	1.111	4.555	0.384	1.209
Dublin	1.620	0.407	1.093	0.189	1.614	0.994	0.839	0.409	0.439		1.725	4.847	1.731	4.163	0.946	1.100	1.652	1.590	0.448	1.119	0.692	1.485	0.818	0.963	0.574	1.886	0.198	0.715
Frankfurt	12.43	2.819	7.136	1.520	14.43	6.057	7.101	3.468	3.207	1.725		19.14	14.88	22.10	7.403	6.421	14.47	16.68	2.789	7.973	6.795	11.16	5.348	12.94	4.792	14.72	1.741	6.887
Glasgow	18.75	4.300	11.16	1.921	18.60	9.615	8.818	3.874	5.028	4.847	19.14		19.76	44.61	9.524	11.09	17.42	18.01	5.646	10.68	7.413	16.18	10.51	9.869	6.824	19.55	1.826	8.422
Hamburg	12.36	2.694	6.372	1.413	18.11	5.373	5.860	3.253	4.580	1.731	14.88	19.76		21.30	5.955	5.951	11.44	12.47	3.177	6.559	6.272	9.920	5.949	7.340	4.432	15.92	1.513	5.085
London	22.60	4.533	12.50	2.088	19.84	11.39	11.96	4.260	5.153	4.163	22.10	44.61	21.30		11.40	12.12	19.84	19.93	5.472	15.33	8.117	17.59	10.43	11.91	7.247	20.56	2.060	9.389
Lyon	5.308	1.638	5.768	0.854	5.877	4.999	3.189	1.622	1.439	0.946	7.403	9.524	5.955	11.40		4.458	10.66	7.327	1.320	5.236	2.868	7.190	2.518	5.301	2.285	7.151	0.979	3.785
Madrid	5.277	1.839	7.535	0.839	5.859	5.634	2.848	1.739	1.505	1.100	6.421	11.09	5.951	12.12	4.458		7.449	6.358	1.493	4.176	2.685	7.022	2.858	3.833	2.266	7.268	0.885	2.882
Milan	9.628	3.349	9.302	1.926	11.75	7.117	5.488	4.038	2.773	1.652	14.47	17.42	11.44	19.84	10.66	7.449		17.38	2.515	7.742	6.133	16.71	4.872	10.28	5.026	14.58	2.364	8.744
Munich	9.991	3.035	7.236	1.795	13.45	5.831	5.392	4.178	2.948	1.590	16.68	18.01	12.47	19.93	7.327	6.358	17.38		2.660	6.800	7.880	12.69	5.211	10.21	5.888	15.35	2.302	7.576
Oslo	2.558	0.659	1.538	0.306	3.017	1.273	1.189	0.654	0.963	0.448	2.789	5.646	3.177	5.472	1.320	1.493	2.515	2.660		1.400	1.187	2.362	2.020	1.423	1.030	3.505	0.297	1.190
Paris	6.611	1.526	4.642	0.781	6.135	4.663	4.659	1.712	1.546	1.119	7.973	10.68	6.559	15.33	5.236	4.176	7.742	6.800	1.400		2.837	6.035	2.611	5.212	2.181	7.182	0.873	3.226
Prague	4.283	1.403	2.997	0.885	8.163	2.429	2.289	2.354	1.510	0.692	6.795	7.413	6.272	8.117	2.868	2.685	6.133	7.880	1.187	2.837		5.217	2.345	3.662	3.226	9.204	1.064	2.497
Rome	8.264	3.903	8.598	2.175	10.30	6.044	4.428	3.983	2.498	1.485	11.16	16.18	9.920	17.59	7.190	7.022	16.71	12.69	2.362	6.035	5.217		4.638	6.853	4.628	13.67	2.393	5.326
Stockholm	4.810	1.311	2.955	0.620	5.946	2.414	2.223	1.314	1.814	0.818	5.348	10.51	5.949	10.43	2.518	2.858	4.872	5.211	2.020	2.611	2.345	4.638		2.707	2.067	7.501	0.581	2.328
Strasbourg	6.376	1.653	4.451	0.928	7.247	3.810	4.015	2.087	1.666	0.963	12.94	9.869	7.340	11.91	5.301	3.833	10.28	10.21	1.423	5.212	3.662	6.853	2.707		2.650	8.229	1.101	5.709
Vienna	3.491	1.218	2.507	0.880	5.043	1.981	1.763	2.619	1.111	0.574	4.792	6.824	4.432	7.247	2.285	2.266	5.026	5.888	1.030	2.181	3.226	4.628	2.067	2.650		6.838	1.098	2.062
Warsaw	10.65	4.281	7.942	2.720	18.99	6.384	5.645	6.909	4.555	1.886	14.72	19.55	15.92	20.56	7.151	7.268	14.58	15.35	3.505	7.182	9.204	13.67	7.501	8.229	6.838		2.662	5.642
Zagreb	1.082	0.580	1.048	0.522	1.691	0.796	0.634	1.136	0.384	0.198	1.741	1.826	1.513	2.060	0.979	0.885	2.364	2.302	0.297	0.873	1.064	2.393	0.581	1.101	1.098	2.662		0.754
Zurich	4.501	1.231	3.316	0.644	5.148	2.717	2.454	1.393	1.209	0.715	6.887	8.422	5.085	9.389	3.785	2.882	8.744	7.576	1.190	3.226	2.497	5.326	2.328	5.709	2.062	5.642	0.754	

Table 3.3.13: Europe Total traffic matrix (Gbit/s) (Gerlach et al., 2004)

3.4. SIMULATION PROCESS

During the Simulation, the following steps and procedures were followed;

After determining the maximum link load, it was then updated accordingly and the process was repeated until the optimum results were obtained .The total traffic matrix was simulated first after which the traffic was separated to determine whether the algorithm is traffic dependant or not. To determine the maximum, minimum and average link load for each traffic matrix, the following steps were executed using the Akita simulation tool:

- lugodo@lugodo-Aspire-E1-531:~$ cd Desktop/
- lugodo@lugodo-Aspire-E1-531:~/Desktop$ cd akita12v2/
- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2$ cd germany
- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2/germany$ cd createRi
- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2/germany/createRi$ make make: Nothing to be done for `all'.
- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2/germany/createRi$ cd
- lugodo@lugodo-Aspire-E1-531:~$ cd Desktop/
- lugodo@lugodo-Aspire-E1-531:~/Desktop$ cd akita12v2/
- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2$ cd germany
- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2/germany$ ./createRi/createRi ./data/germany.csv out.ri

- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2/germany$ perl measure_link_usage.pl ./data/germany_voice_tm_grv.csv ./data/germany.csv out.ri
- lugodo@lugodo-Aspire-E1-531:~/Documents/akita12v2$ ./test.sh
- $init = data/germany.csv
- $fin = data/germany-test.csv
- $tm = data/tm_grv.csv
- $dir = out
- $index = 4
- $factorial=24
- open file ./out/0/
- hello max util = 0.1829952
- hello max util = 0.1829952
- hello max util = 0.1626904 ❖ hello max util = 0.1626904

The above steps apply for the Germany network; the following steps were applied for the pan European network;

- lugodo@lugodo-Aspire-E1-531:~$ cd Desktop/
- lugodo@lugodo-Aspire-E1-531:~/Desktop$ cd akita12v2/

- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2$ cd europe
- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2/ europe $ cd createRi
- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2/ europe /createRi$ make
  make: Nothing to be done for `all'.
- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2/ europe /createRi$ cd
- lugodo@lugodo-Aspire-E1-531:~$ cd Desktop/
- lugodo@lugodo-Aspire-E1-531:~/Desktop$ cd akita12v2/
- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2$ cd europe
- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2/ europe $ ./createRi/createRi ./data/ europe.csv out.ri
- lugodo@lugodo-Aspire-E1-531:~/Desktop/akita12v2/ europe $ perl measure_link_usage.pl ./data/ europe _voice_tm_grv.csv ./data/ europe.csv out.ri
- lugodo@lugodo-Aspire-E1-531:~/Documents/akita12v2/createTransition$ cd ..
- lugodo@lugodo-Aspire-E1-531:~/Documents/akita12v2$ ./test.sh
- $init = data/european.csv
- $fin = data/european-test.csv
- $tm = data/tm_grv.csv
- $dir = out
- $index = 4
- $factorial=24
- open file ./out/0/
- hello max util = 0.0781368
- hello max util = 0.0658008
- hello max util = 0.0658008

ALGORITHM

The algorithm used for the simulations is shown in figure 3.3.3 below;

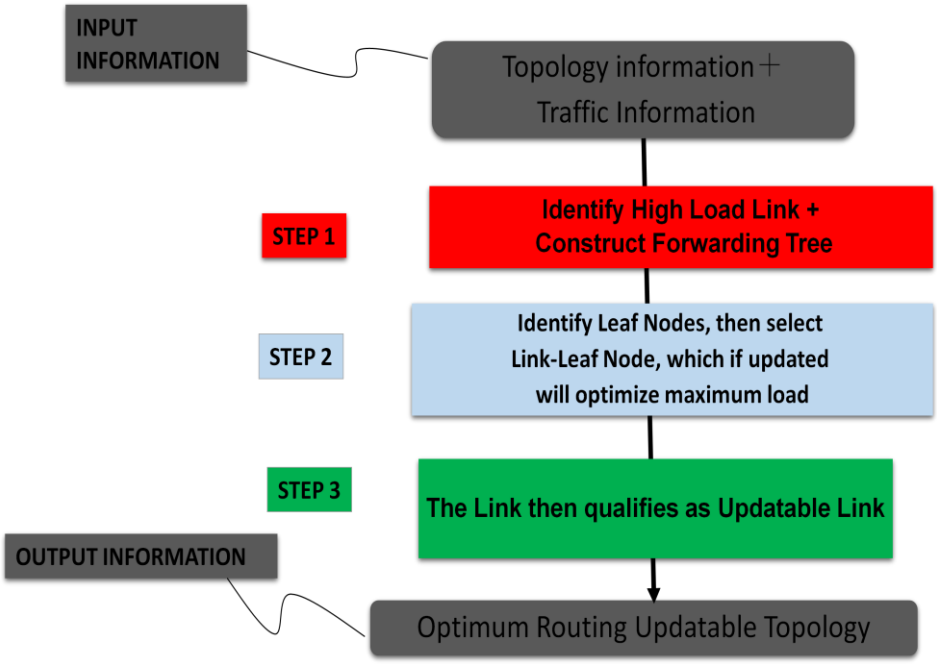

Figure 3.3.3: ULLN – Updatable Link Leaf Node Algorithm (Tembo, 2013)

The following steps were followed to conduct the simulations;

1. Input the information of the network topology i.e. initial & target link costs together with information of the actual traffic matrix. In my case the actual traffic was the total traffic for Germany and Europe. The isolated traffic i.e. actual Data, Voice and IP was also used.
2. Identify the high load link i.e. the link with the maximum load in the network. Then using the destination node, a forwarding tree is constructed.
3. From the forwarding tree constructed, leaf nodes and updatable links connected to the leaf nodes are identified .The updatable links are the ULLN – updatable link leaf nodes. The value of the ULLN is verified i.e. ULLN > 1, otherwise we proceed to update the only identified updatable link. If the ULLN > 1,we determine which updatable link once updated gives the lowest maximum load. An example of a forwarding tree is shown in figure 3.3.4 below. The routing information used for making the forwarding trees are in the appendix.

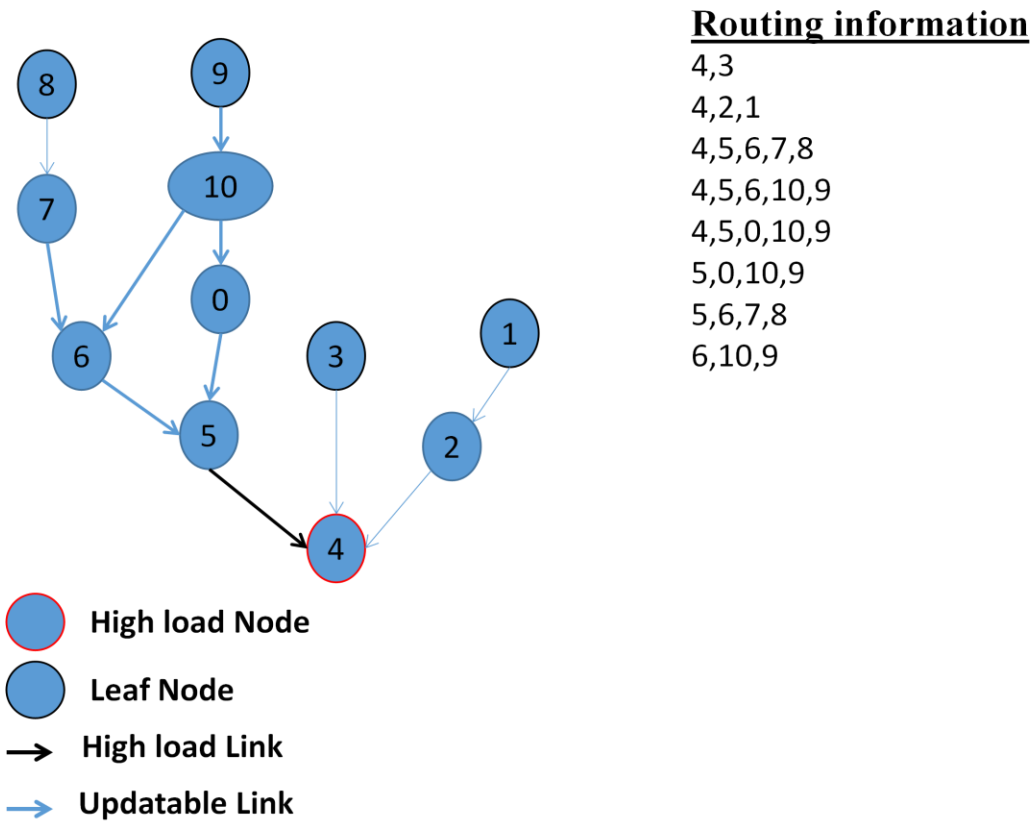

Figure 3.3.4: An example of a forwarding tree

4. We verify if there are any other updatable links to be updated within the topology matrix .If there are more updatable links, we continue with step 1 through step 4 otherwise we exit and confirm the link cost update sequence order.

3.5. CONCLUSION

Using the simulated results, the simulation results /data were collected and analyzed as illustrated under results and discussions. Three scenarios were observed namely;

- congestion – this where congestion was observed
- No Congestion – when there was no congestion but the results were still not optimum, some link overflows
- Optimum – in this scenario there was no congestion observed, optimum routing was achieved i.e. 6 links were obtained providing us with zero congestion

CHAPTER 4: RESULTS AND DISCUSSIONS

4.1. INTRODUCTION

This chapter presents the results and discussions simulations from the previous chapter. The results collected from the simulations are analyzed and discussed in details with the main focus been on the achievement of optimum routing for both network topologies.

4.2. 1st LINK COST UPDATE

Using Akita12v2 for the simulations, brute force approach was used to determine the update sequence order for all the permutations. The input data involves the network topology with known initial link costs and known target link costs. The traffic matrix was also added.

After obtaining optimum results (Optimum routing), the algorithm was tested by separating the traffic, as the results obtained were for the total traffic. Total traffic is simply a summation of all the 3 types (IP, Data & Voice) of traffic as shown in formula 4.10. Traffic was separated and optimum results were obtained for individual traffic (Data, mobile and voice).so the algorithm is not traffic dependant. Total traffic is summation of all the traffic in the network.

$$\text{Total traffic} = \text{IP traffic} + \text{Data traffic} + \text{Voice traffic} \quad (4.10)$$

4.3. RESULTS

The results of the simulations are discussed in the sections below. In this section, we intend to give some detailed explanations and interpretations of the obtained results.

4.3.1. PAN EUROPEAN NETWORK

After the first update, the results show congestion as seen in figure 4.3.1, all links are having zero optimum routing i.e. all 24 links have congestion. Congestion occurs when the transient maximum load exceeds that of the initial state. The idea is to correctly identify links for link cost update that can minimize the maximum load. A forwarding tree is constructed to select links connected to the leaf nodes for link cost update.

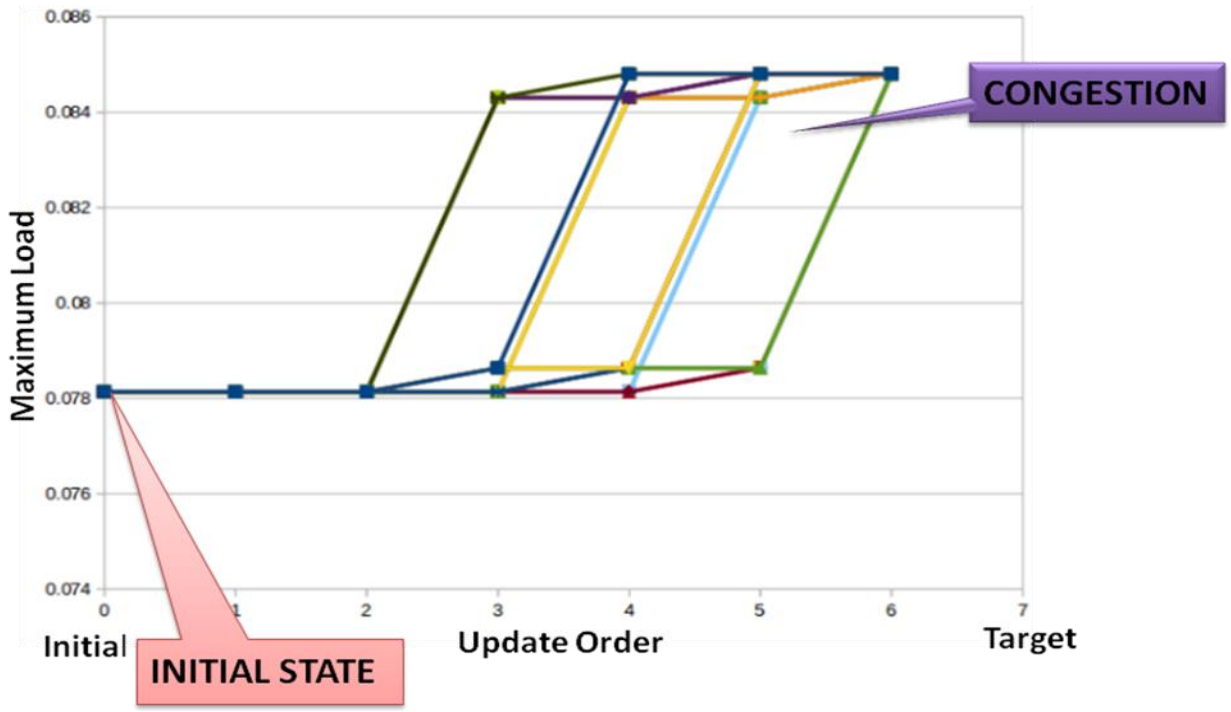

Figure 4.3.1: Congestion after the 1st LCU for Pan European Network

After a number of updates, it can be seen in figure 4.3.2 that optimum routing is achieved for the pan European network, with all links having zero congestion.

Figure 4.3.2:100% optimum routing achieved i.e. All patterns are below the overflow threshold

To verify if the algorithm is traffic dependent, the traffic was isolated i.e. the total traffic was a summation of data, IP and Voice traffic combined. After achieving optimum routing for the Pan European network, traffic was isolated and the results are shown in figures 4.3.3 for Data Traffic isolation, figure 4.3.4 for IP traffic isolation and Figure 4.3.5 for Voice traffic isolation.

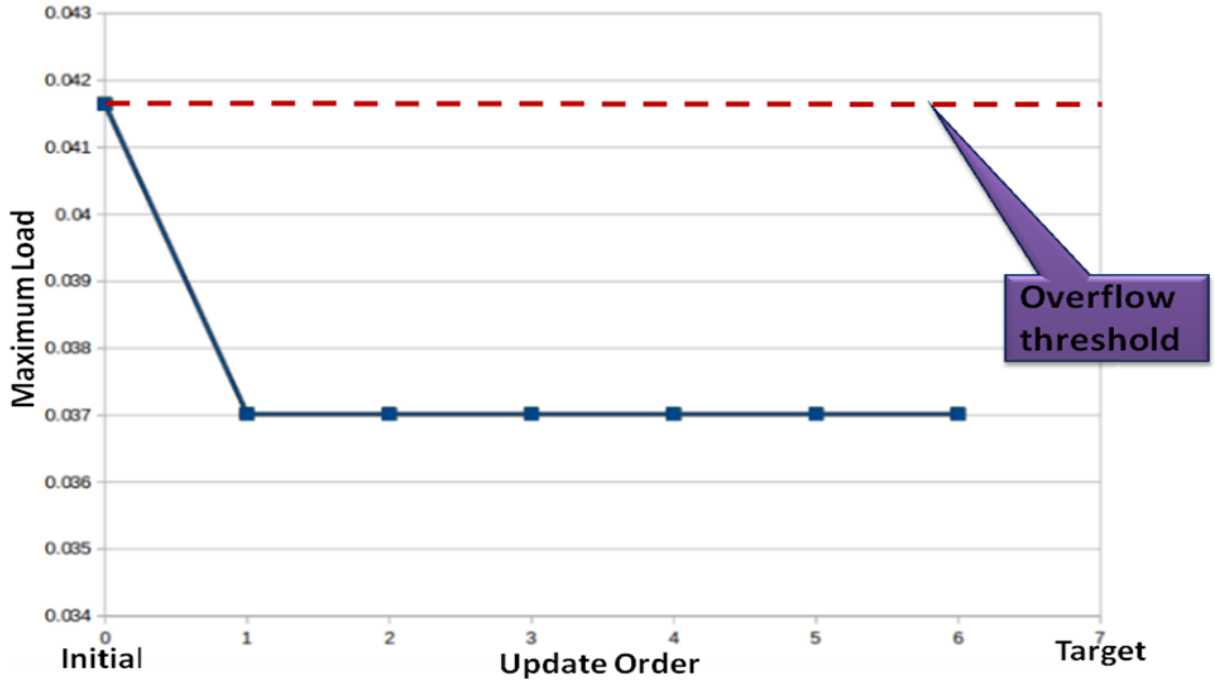

Figure 4.3.3: Traffic isolation results for Data on the Pan European Network

Figure 4.3.4: Traffic Isolation results for IP Traffic _ Pan European Network

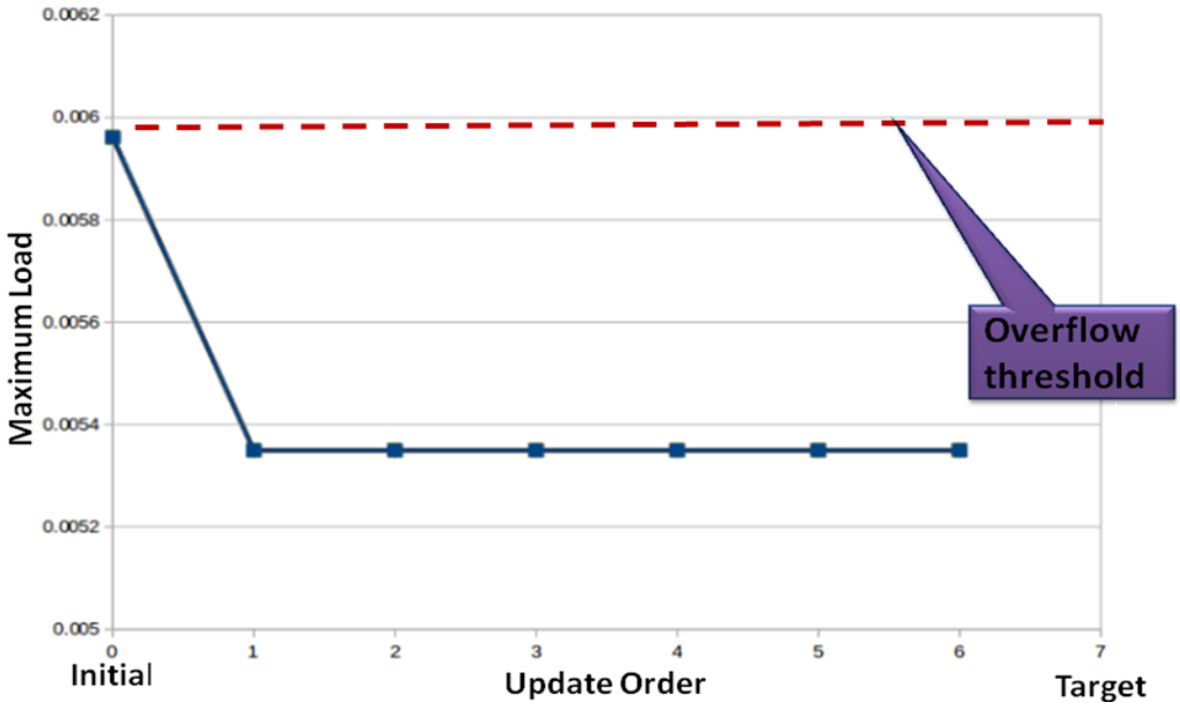

Figure 4.3.5: Traffic Isolation results for Voice Traffic _ Pan European Network

The results of the traffic isolation show that for the pan European network model, the algorithm was not traffic dependent. However this is inconclusive as the algorithm needs to be applied to various traffic models from various sources. The figure 4.3.6 below shows the maximum link loads for the separated traffic modes i.e. Data, IP, Voice and Total Traffic for the Pan European model. Table 4.3.1 gives a summary of the number of links, nodes, updatable nodes and the number of permutations required per update for each network respectively.

Topology	Nodes	Links	Updatable Links	Permutations
European	28	41	6	720
Germany	17	26	6	720

Table 4.3.1: Permutation results for given updatable links

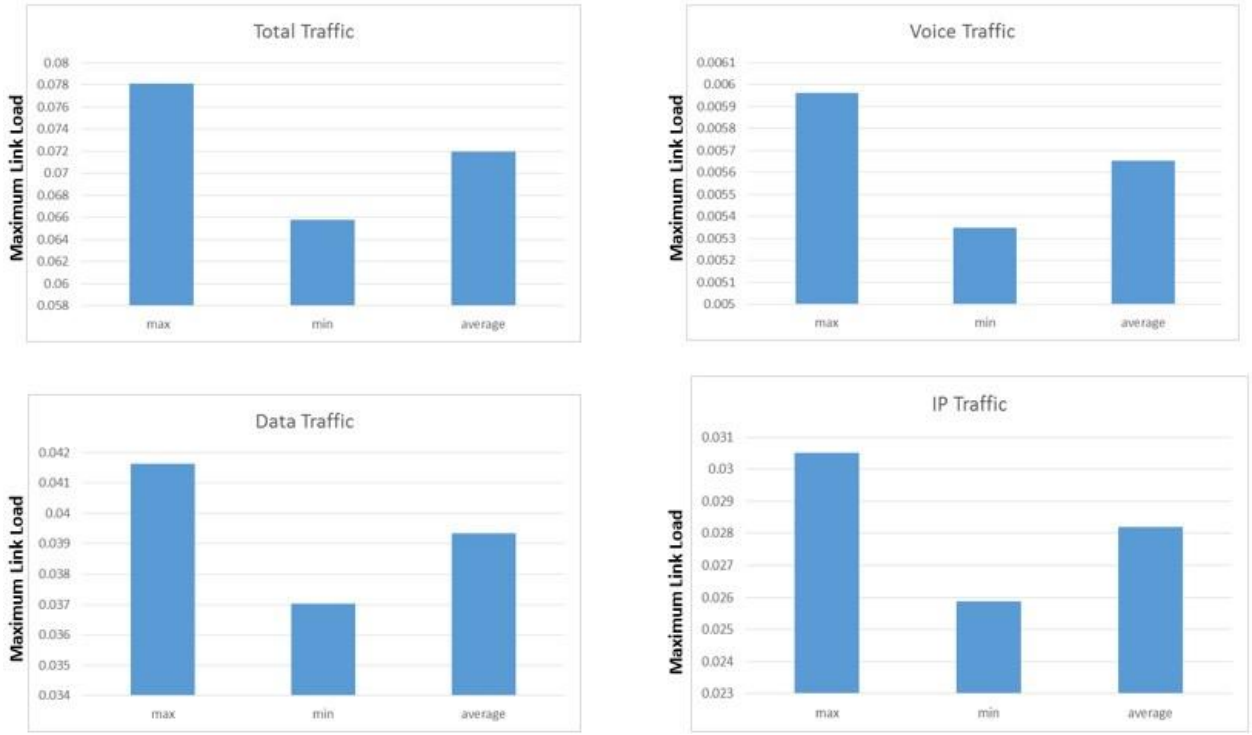

Figure 4.3.6: Maximum Link Loads IP, Data, Voice and Total Traffic

The figure 4.3.7 shows the link utilization for the ideal Total traffic scenario for the Pan European network

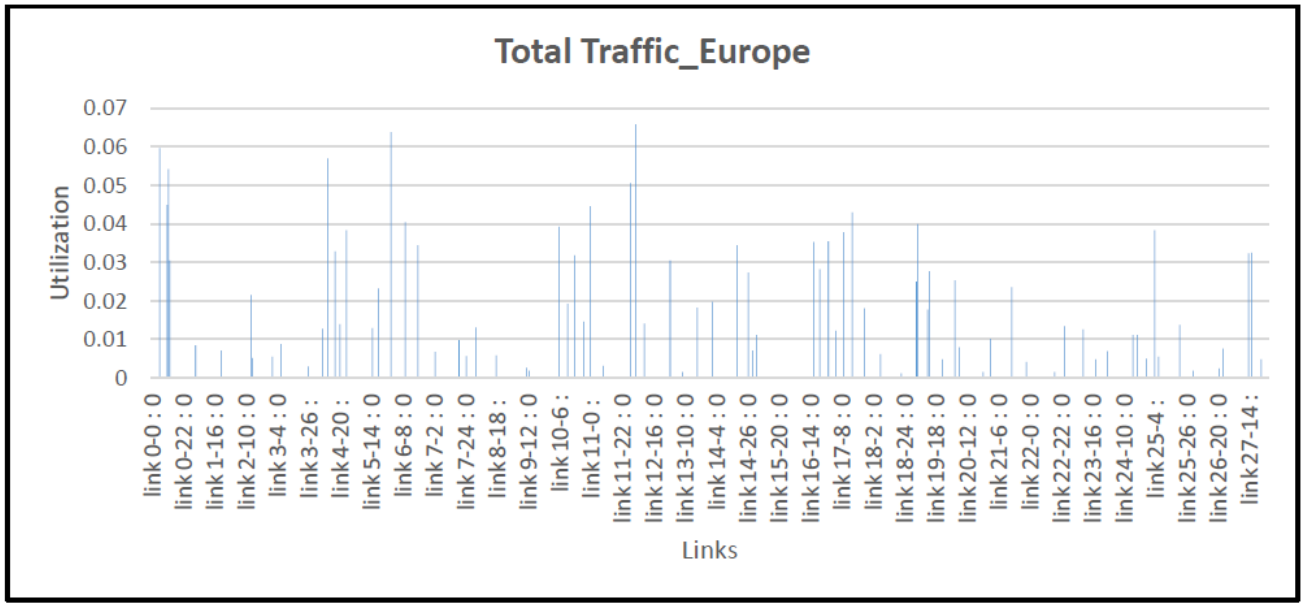

Figure 4.3.7 Total traffic Link Utilization Europe

4.3.2. GERMANY NETWORK

The simulation results for the Germany network are discussed in this section.
After the first LCU, Congestion was observed in the results as shown in figure 4.3.8.

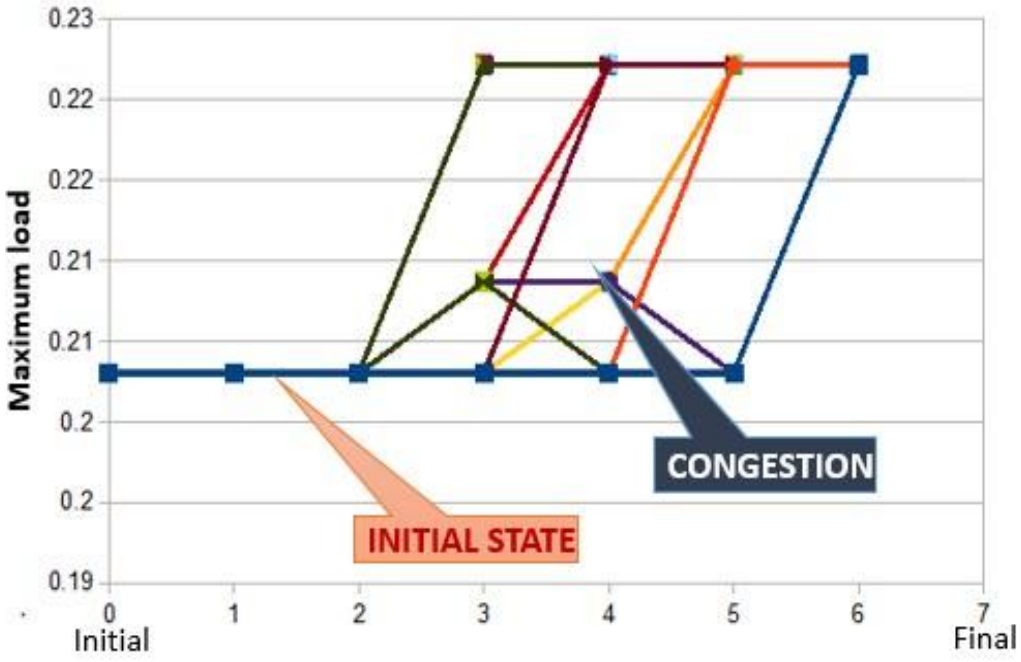

Figure 4.3.8: 100% Congestion - Impossible to achieve optimum routing in this setup (1st LCU)

Several updates were performed until there was zero congestion was achieved. The results can be seen in figure 4.3.9.

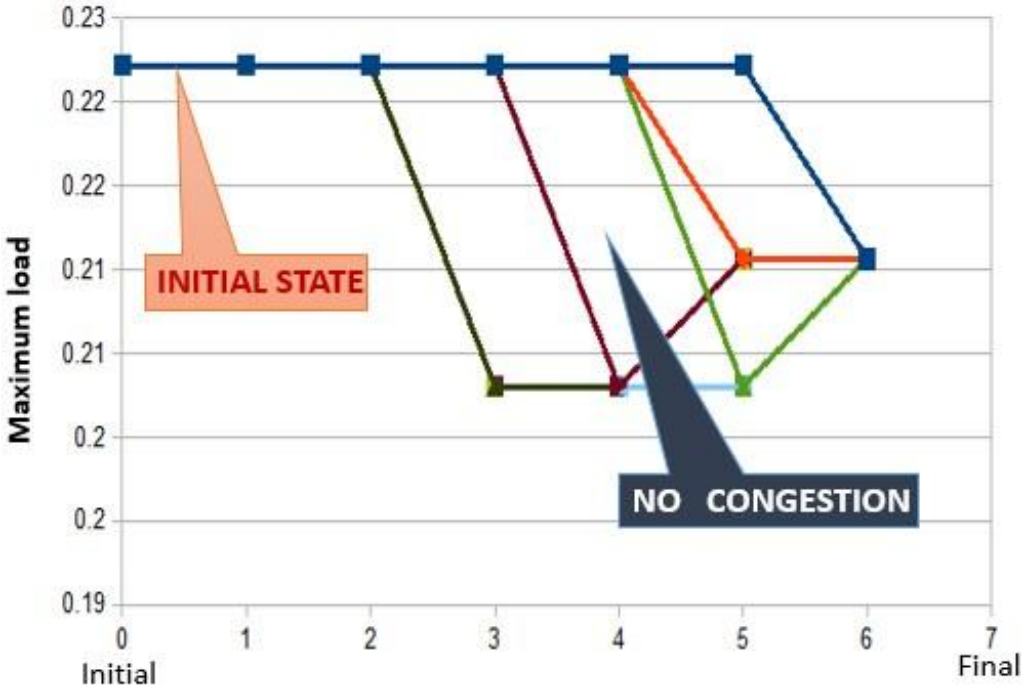

Figure 4.3.9: No congestion _ Not Optimum

Further simulations and updates were performed until optimum routing was achieved. This is shown in Figure 4.3.10.

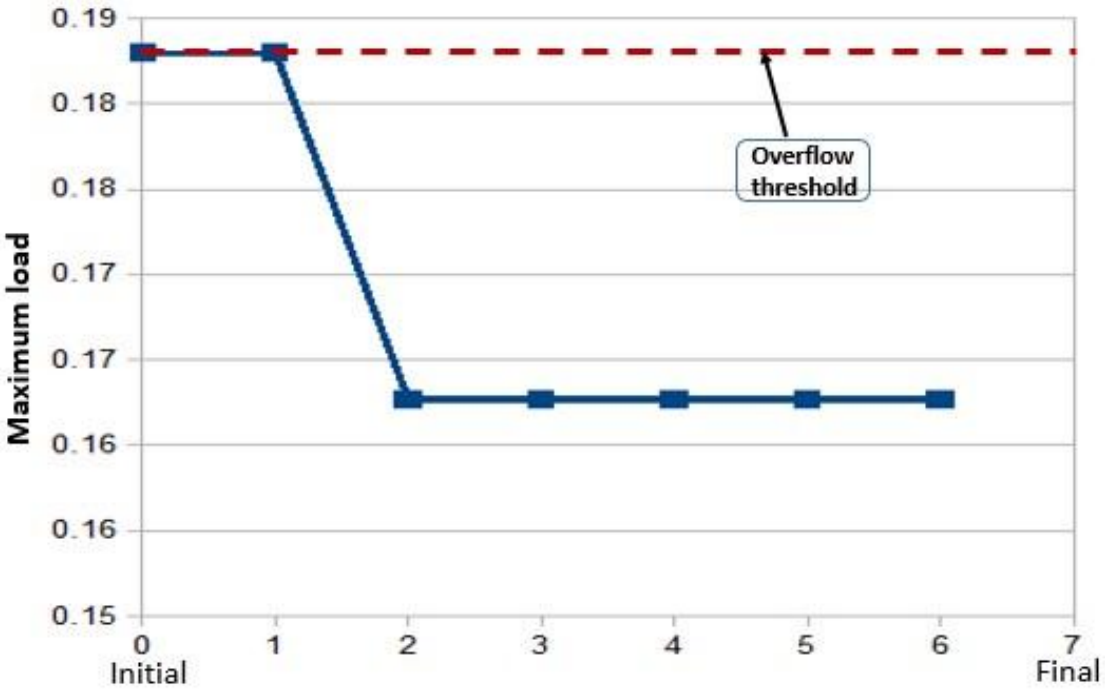

Figure 4.3.10: Optimum Routing – 0% Congestion as all links give optimum routing

Traffic was separated to test the dependency of the algorithm on isolated traffic i.e. Data, Voice and IP. The results are shown in figures 4.3.11, 4.3.12 and 4.3.13. Only Data traffic (figure 4.3.11) was able to achieve optimum traffic. This goes on to show that the algorithm is in fact Traffic dependent. More simulations were required for each traffic type to obtain optimum routing. Figure 4.3.14 shows the link utilization for the total traffic. However for this dissertation the main objective was to achieve optimum routing for the Total Traffic.

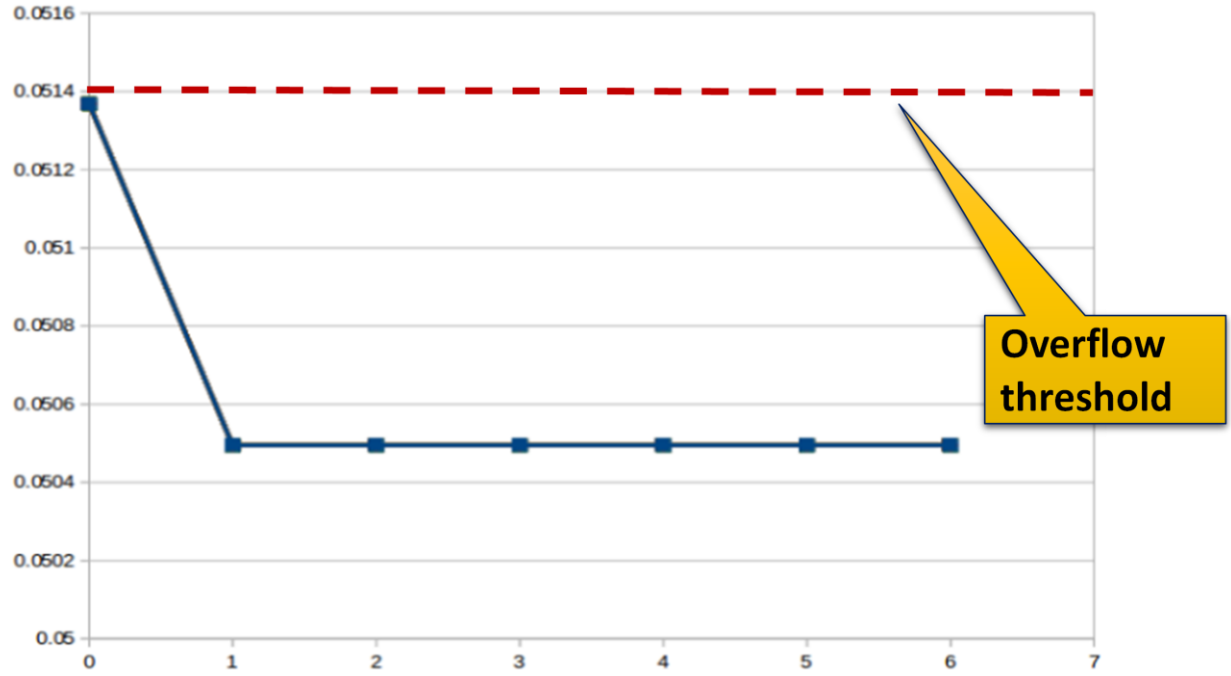

Figure 4.3.11: Data Traffic isolation_ Germany

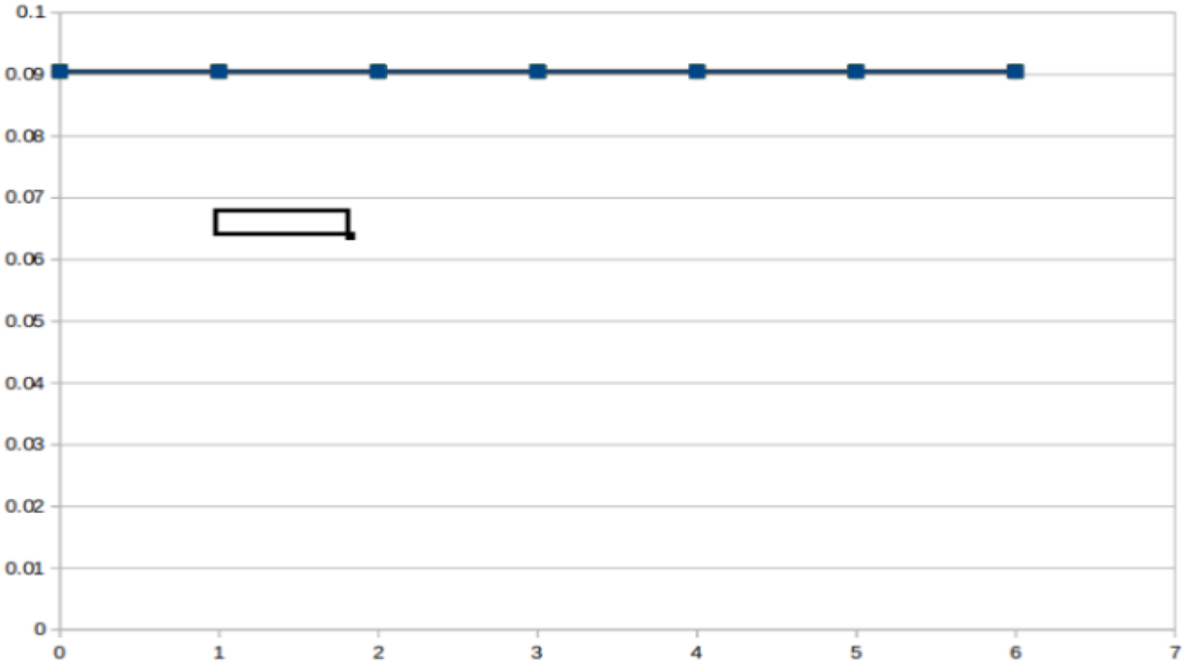

Figure 4.3.12: IP Traffic isolation_ Germany

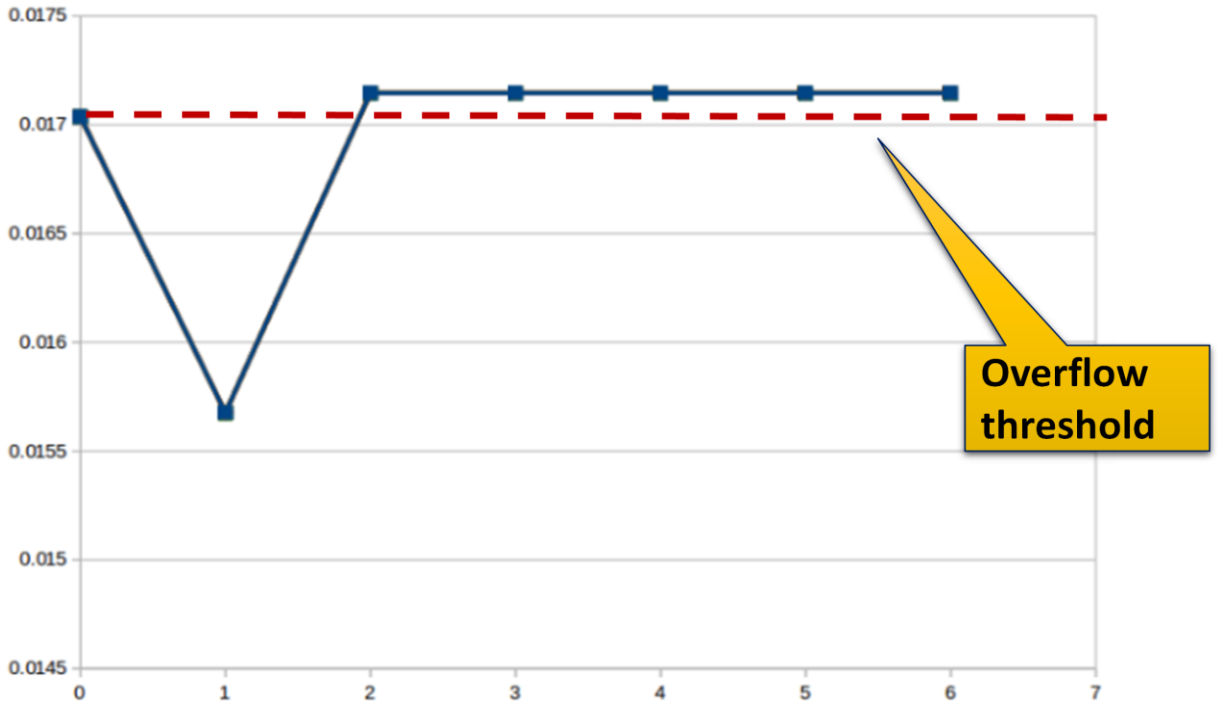

Figure 4.3.13: Voice Traffic isolation_ Germany

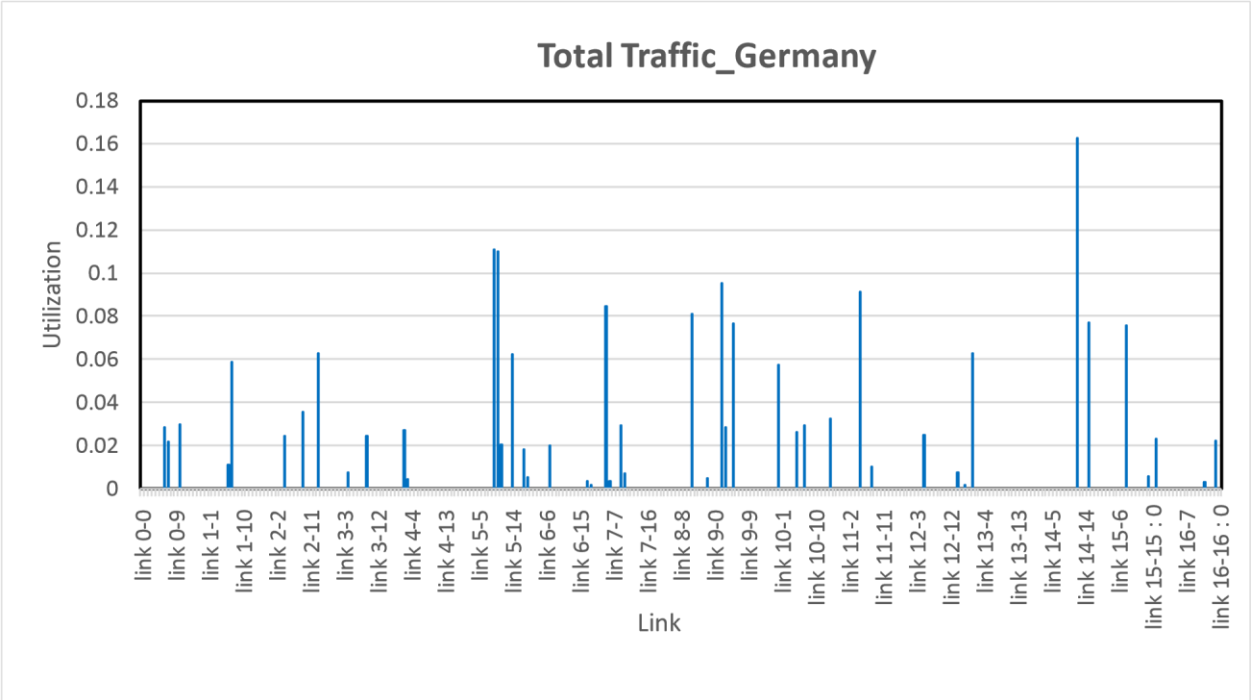

Figure 4.3.14: Link Utilization _ Total Traffic Germany

4.4. DISCUSSION

The results show that optimum routing achieved for both networks for the total traffic matrices. it was also noted that for all traffic matrices, during the 1st LCU, congestion was observed. Congestion occurs when the transient maximum load exceeds that of the initial state. However when traffic was isolated and simulated, the results were inconclusive. For the Pan European network, it was observed that all the 3 types of traffic matrices gave optimum routing after the simulation. Optimum routing is when there is a minimum maximum load in the Network. However for the Germany network, the results were inconclusive as only the Data Traffic matrix resulted in optimum routing. More simulations were required to obtain optimum routing for the IP and Voice traffic respectively. It can therefore be concluded that the algorithm is indeed Traffic dependent. The results for optimum routing i.e. total traffic for both networks, show that we have no link overflows and all the patterns give optimum routing conditions. Therefore the optimum routing performance of 100% is achieved. Optimum routing is defined as achieving the minimum maximum link utilization during the transient period of link cost updates (Tembo 'et al', 2012).

CHAPTER 5:CONCLUSION AND FUTURE WORK

5.1. INTRODUCTION

In this chapter, a summary of the research work and discuss the possible future works to enhance our works is given. The related works, differences in approach and results are also discussed in comparison to our research.

5.2. SUMMARY

A priority link cost update sequence order has been presented to perform transient link cost updates when several links need to be updated for traffic engineering purposes. To find the best solution to this problem is not a practical computation time. We proposed an orderly update of links costs without the need to perform any practical complex permutation computations. The Priority Link Cost Update Sequence Order algorithm can be used to solve such practical problems in computation time. Hence the need of a full search is avoided as we achieve our objective of avoiding transient link overflows and subsequent congestion during transient link cost updates. From the results it has been illustrated that using our algorithm on the given topologies, there exists a 6 links which provide optimum routing during a planned failure. This is useful to ISPs as it assists in traffic engineering related works as traffic can be prioritized accordingly to take advantage of the 6 links providing optimum routing conditions. Congestion can therefore be avoided whenever routine maintenance works are carried out in the network. For the Pan European and Germany network, 6 links giving optimum routing have been identified using our algorithm. The algorithm can be used on any network model and produce optimum routing if applied correctly. The research questions can be summarized as follows;

1. **How does the proposed method avoid congestion?**

   Congestion is avoided at each stage by identifying high load nodes/links and inducing traffic splitting .Prior works have been centered on link rescheduling (Raza et al., 2011),loop avoidance schemes(Fortz et al.,2000).Dionysus and ATOMIP focus on the update order with emphasis on the update time. SWAN (Liu et al.,2013) introduce a sequence of intermediate configurations which results in link capacity wastage as to ensure such a sequence exists requires parts of the link capacities to be left vacant. The ULLN method is a simpler method of identifying high load nodes/links and inducing traffic splitting until the link capacities are at a minimum o within the link capacities.

2. **How does the induction of traffic splitting keep the maximum load to a minimum level or within the link capacities?**

   During the simulations, high load nodes were identified and the updatable leaf nodes attached to that particular high load node were updated to split the traffic. Each resulted in traffic splitting. The key was the establishment of how many updatable links are connected to the leaf node in the forwarding tree.

3. **Is optimum routing i.e. zero congestion achieved?**

   As shown in figures, optimum routing was achieved for both the pan European networks and the Germany network. Optimum routing is achieving the minimum maximum link utilization during the transient period of link cost updates. All patterns are below the overflow threshold.

4. **What is the research contribution?**

   We have presented a priority link cost update sequence order as a new method to perform transient link cost updates when several different links need to be updated for traffic engineering purposes. The algorithm presented eliminates the need for a full search. Each link cost update helps search for a high load node leading to traffic splitting. The optimum routing gives us 6 optimum links in any given network which can be used during planned works. Using this method, we were able to show that using an appropriate link cost update sequence order of link cost for updatable links minimizes transient link overflows (Kamrul and Oki, 2011).

5.3. RECOMMENDATIONS AND FUTURE WORK

We have shown that by using our proposed algorithm on the given network model, we can achieve optimum routing thereby avoiding congestion in IP networks. We have shown that there exists a number of links in the network such that when orderly and sequentially updated, optimum routing is achieved. Using our algorithm, we were able to simulate and identify 6 links for the pan European and Germany network models which when updated gives 100% optimum routing with no link overflows. Our method uses a link cost update sequence order method to update link costs to avoid link overflows with a view to achieve optimum routing conditions. Our simulation results show that the proposed scheme is effective in reducing the maximum link load in the network after failures. Most works have been centered on loop avoidance during the convergence period with very few works looking at congestion avoidance during planned maintenance routines. Our works provide an opportunity to researchers to take an interest and investigate congestion avoidance schemes deeply. Future works will endeavor to apply the algorithm on larger networks and with heavy traffic loads. All network models have their limitations. It would be important to test the algorithm on some local network models with actual traffic. The new SDN paradigm is another interesting area which would be a good platform to test the algorithm.

REFERENCES

Betker, A., Gerlach, C., Hülsermann, R., Jäger, M., Barry, M., Bodamer, S., Späth, J., Gauger, C. and Köhn, M., 2003. Reference transport network scenarios. *MultiTeraNet Report, July*.

Bonaventure, O., 2011. Computer Networking: Principles, Protocols and Practice, Saylor Foundation.

Clark, M.P., 2003. Data Networks, IP and the Internet: Protocols, design and operation, Wiley and Sons.

Dally, W.J. and Towles, B.P., 2004. *Principles and practices of interconnection networks*. Elsevier.

Fortz, B. and Thorup, M., 2000. Internet traffic engineering by optimizing OSPF weights. In *INFOCOM 2000. Nineteenth annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE* (Vol. 2, pp. 519-528). IEEE.

Fortz, B., Rexford, J. and Thorup, M., 2002. Traffic engineering with traditional IP routing protocols. *IEEE Communications Magazine*, *40*(10), pp.118-124.

Francois, P., Filsfils, C., Evans, J. and Bonaventure, O., 2005. Achieving sub-second IGP convergence in large IP networks. *ACM SIGCOMM Computer Communication Review*, *35*(3), pp.35-44.

Francois, P., Shand, M. and Bonaventure, O., 2007, May. Disruption free topology reconfiguration in OSPF networks. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications* (pp. 89-97). IEEE.

Fu, J., Sjodin, P. and Karlsson, G., 2008. Loop-free updates of forwarding tables. *IEEE Transactions on Network and Service Management*, *5*(1), pp.22-35

Hong, C.Y., Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M. and Wattenhofer, R., 2013, August. Achieving high utilization with software-driven WAN. In *ACM SIGCOMM Computer Communication Review* (Vol. 43, No. 4, pp. 15-26). ACM.

Hunt,C., 1999. TCP/IP Network Administration,O'Reilly.

Iannaccone, G., Chuah, C.N., Bhattacharyya, S. and Diot, C., 2004. Feasibility of IP restoration in a tier 1 backbone. *Ieee Network*, *18*(2), pp.13-19.

Jain, R., Ramakrishnan, K., and Chiu, D.M., .1998. Congestion avoidance in computer networks with a connectionless network layer . arXiv preprint cs/9809094.

Jin, X., Liu, H.H., Gandhi, R., Kandula, S., Mahajan, R., Zhang, M., Rexford, J. and Wattenhofer, R., 2014, August. Dynamic scheduling of network updates. In *ACM SIGCOMM Computer Communication Review* (Vol. 44, No. 4, pp. 539-550). ACM.

Jin, X., Liu, H.H., Gandhi, R., Kandula, S., Mahajan, R., Zhang, M., Rexford, J. and Wattenhofer, R., 2014, August. Dynamic scheduling of network updates. In *ACM SIGCOMM Computer Communication Review* (Vol. 44, No. 4, pp. 539-550). ACM.

Kamrul, I.M. and Oki, E., 2011, June. Optimization of OSPF link weight to minimize worst-case network congestion against single-link failure. In *2011 IEEE International Conference on Communications (ICC)* (pp. 1-5). IEEE.

Katz, D. and Ward, D., 2007. BFD for IPv4 and IPv6 (Single Hop)(draft-ietf-bfd-v4v6-1hop-03. txt. *Network Working Group Internet Draft*.

Larry, L.P. and S.B. Bruce., 2003.Computer Networks: A systems Approach, 3rd Edition, Morgan Kaufman Press.

Luo, L., Yu, H., Luo, S. and Zhang, M., 2015, June. Fast lossless traffic migration for SDN updates. In *2015 IEEE International Conference on Communications (ICC)* (pp. 5803-5808). IEEE.

Markopoulou, A., Iannaccone, G., Bhattacharyya, S., Chuah, C.N. and Diot, C., 2004, March. Characterization of failures in an IP backbone. In*INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies* (Vol. 4, pp. 2307-2317). IEEE.

McClurg, J., Hojjat, H., Černý, P. and Foster, N., 2015, June. Efficient synthesis of network updates. In *ACM SIGPLAN Notices* (Vol. 50, No. 6, pp. 196-207). ACM.

Moy, J., 1991. Request for Comments: 1247. Internet Engineering Task Force.

Oran, D., 1990. OSI IS-IS intra-domain routing protocol.

Raj, A.,and Ibe, O.C.,2007. A survey of IP and multiprotocol label switching fast reroute schemes. Computer networks (vol. 51, no. 8, pp. 1882-1907)

Raza, S., Zhu, Y. and Chuah, C.N., 2009, April. Graceful network operations. In *INFOCOM 2009, IEEE* (pp. 289-297). IEEE.

Raza, S., Zhu, Y. and Chuah, C.N., 2011. Graceful network state migrations.*IEEE/ACM Transactions on Networking (TON)*, *19*(4), pp.1097-1110.

Reitblatt, M., Foster, N., Rexford, J. and Walker, D., 2011, November. Consistent updates for software-defined networks: Change you can believe in!. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks* (p. 7). ACM.

Reitblatt, M., Foster, N., Rexford, J., Schlesinger, C. and Walker, D., 2012, August. Abstractions for network update. In Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication (pp. 323-334). ACM.

Shi, L., Fu, J. and Fu, X., 2009, June. Loop-free forwarding table updates with minimal link overflow. In *2009 IEEE International Conference on Communications* (pp. 1-6). IEEE.

Teixeira, R. and Rexford, J., 2006. Managing routing disruptions in internet service provider networks. *IEEE Communications Magazine*, *44*(3), pp.160-165.

Tembo S.,2013. *A study on new congestion avoidance methods in IP routing protocols*. PhD Thesis, Akita University.

Tembo,S., Yukimatsu,K., Takahashi,R., Kamamura,S., Miyamura,T., Shiomoto,K. 2012."A New Backup Topology Design Method for Congestion Avoidance in IP Fast Reroute," International Journal of Networks and Communications, 2(5): 123-131.

Vanbever, L., Vissicchio, S., Pelsser, C., Francois, P. and Bonaventure, O., 2011, August. Seamless network-wide IGP migrations. In *ACM SIGCOMM Computer Communication Review* (Vol. 41, No. 4, pp. 314-325). ACM.

Verma, D.C., 2004. Service level agreements on IP networks. *Proceedings of the IEEE*, *92*(9), pp.1382-1388.
Wiley.

Zheng, J., Xu, H., Chen, G. and Dai, H., 2015, November. Minimizing transient congestion during network update in data centers. In *2015 IEEE 23rd International Conference on Network Protocols (ICNP)* (pp. 1-10). IEEE.

APPENDIX

Source code for the Algorithm

```
#!/usr/bin/perl
# mrcResEvalv3.pl
# usage : %./autoEval.pl <initial topology> <final topology> <tm> <directory> <index>
# <initial topology> :
# <final topology> :
# <tm> : traffic matrix
# <directory> : all sets of link cost (e.g. out/ directory)
# <index> : number of transient patterns
# open input file
$init = "$ARGV[0]";
$fin = "$ARGV[1]";
$tm = "$ARGV[2]";
$dir = "$ARGV[3]";
$index = "$ARGV[4]";
print "\$init = $init\n";
print "\$fin = $fin\n";
print "\$tm = $tm\n";
print "\$dir = $dir\n";
print "\$index = $index\n";
# create loop num
$factorial = 1;
$n = $index;
while ($n > 1) {
$factorial = $factorial * $n;
$n = $n - 1;
}
print "\$factorial=$factorial\n";
@mu_matrix = (); #

for ($i = 0; $i < $factorial; $i++) {
@files = &GetFilelist("./$dir/$i/", csv, 0) or die "open mrc file error";
print "open file ./$dir/$i/\n";
$nums = @files;
for ($j = 0; $j < $nums; $j++) { #
        # RI
        my $ret = `./createRi/createRi ./$dir/$i/$j.csv out.ri`;
        my $ret2 = `perl measure_link_usage.pl $tm $init out.ri`;
        open MU, "max_usage.dat" or die "open error max_usage.dat";
        $mu = <MU>;
        chomp ($mu);
        print "hello max util = $mu\n";
```

```
        $mu_matrix[$i][$j] = $mu;
        close (MU);

#               exit;

}
}

# print matrix
open OUT, ">result.csv" or die "open error result.csv";
print OUT ",";
for ($j = 0; $j < $factorial; $j++) {
print OUT "$j,";
}
print OUT "\n";
for ($j = 0; $j < $nums; $j++) {
print OUT "$j,";
for ($i = 0; $i < $factorial; $i++) {
        print OUT "$mu_matrix[$i][$j],";
}
print OUT "\n";
}
```

Raw Data for figures 3.3.6 and 3.3.5

EUROPE

link 0-0 : 0	0
link 0-1 : 0	0
link 0-2 : 0	0
link 0-3 : 0	0
link 0-4 : 0	0
link 0-5 : 0	0
link 0-6 :	0.059572
link 0-7 : 0	0
link 0-8 : 0	0
link 0-9 : 0	0
link 0-10 : 0	0
link 0-11 :	0.044914
link 0-12 :	0.054107
link 0-13 :	0.030364
link 0-14 : 0	0
link 0-15 : 0	0
link 0-16 : 0	0
link 0-17 : 0	0
link 0-18 : 0	0
link 0-19 : 0	0
link 0-20 : 0	0
link 0-21 : 0	0
link 0-22 : 0	0
link 0-23 : 0	0
link 0-24 : 0	0
link 0-25 : 0	0
link 0-26 : 0	0
link 0-27 : 0	0
link 1-0 : 0	0
link 1-1 : 0	0
link 1-2 : 0	0
link 1-3 :	0.008466
link 1-4 : 0	0
link 1-5 : 0	0
link 1-6 : 0	0

link 1-7 : 0	0
link 1-8 : 0	0
link 1-9 : 0	0

link 2-22 : 0	0

link 1-10 : 0	0
link 1-11 : 0	0
link 1-12 : 0	0
link 1-13 : 0	0
link 1-14 : 0	0
link 1-15 : 0	0
link 1-16 : 0	0
link 1-17 : 0	0
link 1-18 : 0	0
link 1-19 : 0	0
link 1-20 : 0	0
link 1-21 :	0.007055
link 1-22 : 0	0
link 1-23 : 0	0
link 1-24 : 0	0
link 1-25 : 0	0
link 1-26 : 0	0
link 1-27 : 0	0
link 2-0 : 0	0
link 2-1 : 0	0
link 2-2 : 0	0
link 2-3 : 0	0
link 2-4 : 0	0
link 2-5 : 0	0
link 2-6 : 0	0
link 2-7 : 0	0
link 2-8 : 0	0
link 2-9 : 0	0
link 2-10 : 0	0
link 2-11 : 0	0
link 2-12 : 0	0
link 2-13 : 0	0
link 2-14 :	0.021412
link 2-15 :	0.005226
link 2-16 : 0	0
link 2-17 : 0	0
link 2-18 : 0	0
link 2-19 : 0	0
link 2-20 : 0	0
link 2-21 : 0	0

link 2-23 : 0	0
link 2-24 : 0	0
link 2-25 : 0	0
link 2-26 : 0	0
link 2-27 : 0	0
link 3-0 : 0	0
link 3-1 :	0.005475
link 3-2 : 0	0
link 3-3 : 0	0
link 3-4 : 0	0
link 3-5 : 0	0
link 3-6 : 0	0
link 3-7 :	0.008752
link 3-8 : 0	0
link 3-9 : 0	0
link 3-10 : 0	0
link 3-11 : 0	0
link 3-12 : 0	0
link 3-13 : 0	0
link 3-14 : 0	0
link 3-15 : 0	0
link 3-16 : 0	0
link 3-17 : 0	0
link 3-18 : 0	0
link 3-19 : 0	0
link 3-20 : 0	0
link 3-21 : 0	0
link 3-22 : 0	0
link 3-23 : 0	0
link 3-24 : 0	0
link 3-25 : 0	0
link 3-26 :	0.002828
link 3-27 : 0	0
link 4-0 : 0	0
link 4-1 : 0	0
link 4-2 : 0	0
link 4-3 : 0	0
link 4-4 : 0	0
link 4-5 : 0	0
link 4-6 : 0	0

link 4-8 :	0.012787
link 4-9 : 0	0
link 4-10 : 0	0
link 4-11 : 0	0
link 4-12 :	0.056845
link 4-13 : 0	0
link 4-14 : 0	0
link 4-15 : 0	0
link 4-16 : 0	0
link 4-17 :	0.032721
link 4-18 : 0	0
link 4-19 : 0	0
link 4-20 :	0.013941
link 4-21 : 0	0
link 4-22 : 0	0
link 4-23 : 0	0
link 4-24 : 0	0
link 4-25 :	0.038289
link 4-26 : 0	0
link 4-27 : 0	0
link 5-0 : 0	0
link 5-1 : 0	0
link 5-2 : 0	0
link 5-3 : 0	0
link 5-4 : 0	0
link 5-5 : 0	0
link 5-6 : 0	0
link 5-7 : 0	0
link 5-8 : 0	0
link 5-9 : 0	0
link 5-10 : 0	0
link 5-11 : 0	0
link 5-12 : 0	0
link 5-13 : 0	0
link 5-14 : 0	0
link 5-15 :	0.012903
link 5-16 : 0	0
link 5-17 : 0	0
link 5-18 : 0	0
link 5-19 :	0.023161

link 5-20 : 0	0
link 5-21 : 0	0
link 5-22 : 0	0
link 5-23 : 0	0
link 5-24 : 0	0
link 5-25 : 0	0
link 5-26 : 0	0
link 5-27 : 0	0
link 6-0 :	0.063697
link 6-1 : 0	0
link 6-2 : 0	0
link 6-3 : 0	0
link 6-4 : 0	0
link 6-5 : 0	0
link 6-6 : 0	0
link 6-7 : 0	0
link 6-8 : 0	0
link 6-9 : 0	0
link 6-10 :	0.040438
link 6-11 : 0	0
link 6-12 : 0	0
link 6-13 : 0	0
link 6-14 : 0	0
link 6-15 : 0	0
link 6-16 : 0	0
link 6-17 : 0	0
link 6-18 : 0	0
link 6-19 :	0.034341
link 6-20 : 0	0
link 6-21 : 0	0
link 6-22 : 0	0
link 6-23 : 0	0
link 6-24 : 0	0
link 6-25 : 0	0
link 6-26 : 0	0
link 6-27 : 0	0
link 7-0 : 0	0
link 7-1 : 0	0

link 7-5 : 0	0
link 7-6 : 0	0
link 7-7 : 0	0
link 7-8 : 0	0
link 7-9 : 0	0
link 7-10 : 0	0
link 7-11 : 0	0
link 7-12 : 0	0
link 7-13 : 0	0
link 7-14 : 0	0
link 7-15 : 0	0
link 7-16 : 0	0
link 7-17 : 0	0
link 7-18 : 0	0
link 7-19 : 0	0
link 7-20 :	0.009714
link 7-21 : 0	0
link 7-22 : 0	0
link 7-23 : 0	0
link 7-24 : 0	0
link 7-25 :	0.005731
link 7-26 : 0	0
link 7-27 : 0	0
link 8-0 : 0	0
link 8-1 : 0	0
link 8-2 : 0	0
link 8-3 : 0	0
link 8-4 :	0.013102
link 8-5 : 0	0
link 8-6 : 0	0
link 8-7 : 0	0
link 8-8 : 0	0
link 8-9 : 0	0
link 8-10 : 0	0
link 8-11 : 0	0
link 8-12 : 0	0
link 8-13 : 0	0
link 8-14 : 0	0

link 7-2 : 0	0
link 7-3 :	0.006694
link 7-4 : 0	0

link 8-15 : 0	0
link 8-16 : 0	0
link 8-17 : 0	0

link 8-18 :	0.00588
link 8-19 : 0	0
link 8-20 : 0	0
link 8-21 : 0	0
link 8-22 : 0	0
link 8-23 : 0	0
link 8-24 : 0	0
link 8-25 : 0	0
link 8-26 : 0	0
link 8-27 : 0	0
link 9-0 : 0	0
link 9-1 : 0	0
link 9-2 : 0	0
link 9-3 : 0	0
link 9-4 : 0	0
link 9-5 : 0	0
link 9-6 : 0	0
link 9-7 : 0	0
link 9-8 : 0	0
link 9-9 : 0	0
link 9-10 : 0	0
link 9-11 :	0.002616
link 9-12 : 0	0
link 9-13 :	0.001952
link 9-14 : 0	0
link 9-15 : 0	0
link 9-16 : 0	0
link 9-17 : 0	0
link 9-18 : 0	0
link 9-19 : 0	0
link 9-20 : 0	0
link 9-21 : 0	0
link 9-22 : 0	0
link 9-23 : 0	0
link 9-24 : 0	0
link 9-25 : 0	0
link 9-26 : 0	0
link 9-27 : 0	0

link 10-3 : 0	0
link 10-4 : 0	0
link 10-5 : 0	0
link 10-6 :	0.039096
link 10-7 : 0	0
link 10-8 : 0	0
link 10-9 : 0	0
link 10-10 : 0	0
link 10-11 : 0	0
link 10-12 :	0.019314
link 10-13 : 0	0
link 10-14 : 0	0
link 10-15 : 0	0
link 10-16 : 0	0
link 10-17 :	0.031764
link 10-18 : 0	0
link 10-19 : 0	0
link 10-20 : 0	0
link 10-21 : 0	0
link 10-22 : 0	0
link 10-23 :	0.01464
link 10-24 : 0	0
link 10-25 : 0	0
link 10-26 : 0	0
link 10-27 : 0	0
link 11-0 :	0.044442
link 11-1 : 0	0
link 11-2 : 0	0
link 11-3 : 0	0
link 11-4 : 0	0
link 11-5 : 0	0
link 11-6 : 0	0
link 11-7 : 0	0
link 11-8 : 0	0
link 11-9 :	0.003088
link 11-10 : 0	0
link 11-11 : 0	0
link 11-12 : 0	0

link 10-0 : 0	0
link 10-1 : 0	0
link 10-2 : 0	0

link 11-13 : 0	0
link 11-14 : 0	0
link 11-15 : 0	0

link 11-16 : 0	0
link 11-17 : 0	0
link 11-18 : 0	0
link 11-19 : 0	0
link 11-20 : 0	0
link 11-21 : 0	0
link 11-22 : 0	0
link 11-23 : 0	0
link 11-24 : 0	0
link 11-25 : 0	0
link 11-26 : 0	0
link 11-27 : 0	0
link 12-0 :	0.050455
link 12-1 : 0	0
link 12-2 : 0	0
link 12-3 : 0	0
link 12-4 :	0.065801
link 12-5 : 0	0
link 12-6 : 0	0
link 12-7 : 0	0
link 12-8 : 0	0
link 12-9 : 0	0
link 12-10 :	0.014011
link 12-11 : 0	0
link 12-12 : 0	0
link 12-13 : 0	0
link 12-14 : 0	0
link 12-15 : 0	0
link 12-16 : 0	0
link 12-17 : 0	0
link 12-18 : 0	0
link 12-19 : 0	0
link 12-20 : 0	0
link 12-21 : 0	0
link 12-22 : 0	0
link 12-23 : 0	0
link 12-24 : 0	0
link 12-25 : 0	0

link 13-1 : 0	0
link 13-2 : 0	0
link 13-3 : 0	0
link 13-4 : 0	0
link 13-5 : 0	0
link 13-6 : 0	0
link 13-7 : 0	0
link 13-8 : 0	0
link 13-9 :	0.001479
link 13-10 : 0	0
link 13-11 : 0	0
link 13-12 : 0	0
link 13-13 : 0	0
link 13-14 : 0	0
link 13-15 : 0	0
link 13-16 : 0	0
link 13-17 : 0	0
link 13-18 : 0	0
link 13-19 :	0.018197
link 13-20 : 0	0
link 13-21 : 0	0
link 13-22 : 0	0
link 13-23 : 0	0
link 13-24 : 0	0
link 13-25 : 0	0
link 13-26 : 0	0
link 13-27 : 0	0
link 14-0 : 0	0
link 14-1 : 0	0
link 14-2 :	0.019672
link 14-3 : 0	0
link 14-4 : 0	0
link 14-5 : 0	0
link 14-6 : 0	0
link 14-7 : 0	0
link 14-8 : 0	0
link 14-9 : 0	0
link 14-10 : 0	0

link 12-26 : 0	0
link 12-27 : 0	0
link 13-0 :	0.030364

link 14-11 : 0	0
link 14-12 : 0	0
link 14-13 : 0	0

link 14-14 : 0	0
link 14-15 : 0	0
link 14-16 : 0	0
link 14-17 : 0	0
link 14-18 : 0	0
link 14-19 :	0.03431
link 14-20 : 0	0
link 14-21 : 0	0
link 14-22 : 0	0
link 14-23 : 0	0
link 14-24 : 0	0
link 14-25 : 0	0
link 14-26 : 0	0
link 14-27 :	0.027319
link 15-0 : 0	0
link 15-1 : 0	0
link 15-2 :	0.006966
link 15-3 : 0	0
link 15-4 : 0	0
link 15-5 :	0.011163
link 15-6 : 0	0
link 15-7 : 0	0
link 15-8 : 0	0
link 15-9 : 0	0
link 15-10 : 0	0
link 15-11 : 0	0
link 15-12 : 0	0
link 15-13 : 0	0
link 15-14 : 0	0
link 15-15 : 0	0
link 15-16 : 0	0
link 15-17 : 0	0
link 15-18 : 0	0
link 15-19 : 0	0
link 15-20 : 0	0
link 15-21 : 0	0
link 15-22 : 0	0
link 15-23 : 0	0

link 15-27 : 0	0
link 16-0 : 0	0
link 16-1 : 0	0
link 16-2 : 0	0
link 16-3 : 0	0
link 16-4 : 0	0
link 16-5 : 0	0
link 16-6 : 0	0
link 16-7 : 0	0
link 16-8 : 0	0
link 16-9 : 0	0
link 16-10 : 0	0
link 16-11 : 0	0
link 16-12 : 0	0
link 16-13 : 0	0
link 16-14 : 0	0
link 16-15 : 0	0
link 16-16 : 0	0
link 16-17 :	0.035256
link 16-18 : 0	0
link 16-19 : 0	0
link 16-20 : 0	0
link 16-21 :	0.028173
link 16-22 : 0	0
link 16-23 : 0	0
link 16-24 : 0	0
link 16-25 : 0	0
link 16-26 : 0	0
link 16-27 :	0.035396
link 17-0 : 0	0
link 17-1 : 0	0
link 17-2 : 0	0
link 17-3 : 0	0
link 17-4 :	0.012217
link 17-5 : 0	0
link 17-6 : 0	0
link 17-7 : 0	0
link 17-8 : 0	0

link 15-24 : 0	0
link 15-25 : 0	0
link 15-26 : 0	0

link 17-9 : 0	0
link 17-10 :	0.037776
link 17-11 : 0	0

link 17-12 : 0	0
link 17-13 : 0	0
link 17-14 : 0	0
link 17-15 : 0	0
link 17-16 :	0.042896
link 17-17 : 0	0
link 17-18 : 0	0
link 17-19 : 0	0
link 17-20 : 0	0
link 17-21 : 0	0
link 17-22 : 0	0
link 17-23 : 0	0
link 17-24 :	0.017976
link 17-25 : 0	0
link 17-26 : 0	0
link 17-27 : 0	0
link 18-0 : 0	0
link 18-1 : 0	0
link 18-2 : 0	0
link 18-3 : 0	0
link 18-4 : 0	0
link 18-5 : 0	0
link 18-6 : 0	0
link 18-7 : 0	0
link 18-8 :	0.006195
link 18-9 : 0	0
link 18-10 : 0	0
link 18-11 : 0	0
link 18-12 : 0	0
link 18-13 : 0	0
link 18-14 : 0	0
link 18-15 : 0	0
link 18-16 : 0	0
link 18-17 : 0	0
link 18-18 : 0	0
link 18-19 : 0	0
link 18-20 : 0	0
link 18-21 : 0	0

link 18-25 : 0	0
link 18-26 : 0	0
link 18-27 : 0	0
link 19-0 : 0	0
link 19-1 : 0	0
link 19-2 : 0	0
link 19-3 : 0	0
link 19-4 : 0	0
link 19-5 :	0.024901
link 19-6 :	0.039807
link 19-7 : 0	0
link 19-8 : 0	0
link 19-9 : 0	0
link 19-10 : 0	0
link 19-11 : 0	0
link 19-12 : 0	0
link 19-13 :	0.017724
link 19-14 :	0.027577
link 19-15 : 0	0
link 19-16 : 0	0
link 19-17 : 0	0
link 19-18 : 0	0
link 19-19 : 0	0
link 19-20 : 0	0
link 19-21 : 0	0
link 19-22 : 0	0
link 19-23 :	0.004731
link 19-24 : 0	0
link 19-25 : 0	0
link 19-26 : 0	0
link 19-27 : 0	0
link 20-0 : 0	0
link 20-1 : 0	0
link 20-2 : 0	0
link 20-3 : 0	0
link 20-4 :	0.025175
link 20-5 : 0	0
link 20-6 : 0	0

link 18-22 :	0.001234
link 18-23 : 0	0
link 18-24 : 0	0

link 20-7 :	0.007971
link 20-8 : 0	0
link 20-9 : 0	0

link 20-10 : 0	0
link 20-11 : 0	0
link 20-12 : 0	0
link 20-13 : 0	0
link 20-14 : 0	0
link 20-15 : 0	0
link 20-16 : 0	0
link 20-17 : 0	0
link 20-18 : 0	0
link 20-19 : 0	0
link 20-20 : 0	0
link 20-21 : 0	0
link 20-22 : 0	0
link 20-23 : 0	0
link 20-24 :	0.001617
link 20-25 : 0	0
link 20-26 : 0	0
link 20-27 : 0	0
link 21-0 : 0	0
link 21-1 :	0.010046
link 21-2 : 0	0
link 21-3 : 0	0
link 21-4 : 0	0
link 21-5 : 0	0
link 21-6 : 0	0
link 21-7 : 0	0
link 21-8 : 0	0
link 21-9 : 0	0
link 21-10 : 0	0
link 21-11 : 0	0
link 21-12 : 0	0
link 21-13 : 0	0
link 21-14 : 0	0
link 21-15 : 0	0
link 21-16 :	0.023476
link 21-17 : 0	0
link 21-18 : 0	0
link 21-19 : 0	0

link 21-23 : 0	0
link 21-24 : 0	0
link 21-25 : 0	0
link 21-26 :	0.004142
link 21-27 : 0	0
link 22-0 : 0	0
link 22-1 : 0	0
link 22-2 : 0	0
link 22-3 : 0	0
link 22-4 : 0	0
link 22-5 : 0	0
link 22-6 : 0	0
link 22-7 : 0	0
link 22-8 : 0	0
link 22-9 : 0	0
link 22-10 : 0	0
link 22-11 : 0	0
link 22-12 : 0	0
link 22-13 : 0	0
link 22-14 : 0	0
link 22-15 : 0	0
link 22-16 : 0	0
link 22-17 : 0	0
link 22-18 :	0.001549
link 22-19 : 0	0
link 22-20 : 0	0
link 22-21 : 0	0
link 22-22 : 0	0
link 22-23 : 0	0
link 22-24 : 0	0
link 22-25 :	0.013374
link 22-26 : 0	0
link 22-27 : 0	0
link 23-0 : 0	0
link 23-1 : 0	0
link 23-2 : 0	0
link 23-3 : 0	0
link 23-4 : 0	0

link 21-20 : 0	0
link 21-21 : 0	0
link 21-22 : 0	0

link 23-5 : 0	0
link 23-6 : 0	0
link 23-7 : 0	0

link 23-8 : 0	0
link 23-9 : 0	0
link 23-10 :	0.01259
link 23-11 : 0	0
link 23-12 : 0	0
link 23-13 : 0	0
link 23-14 : 0	0
link 23-15 : 0	0
link 23-16 : 0	0
link 23-17 : 0	0
link 23-18 : 0	0
link 23-19 :	0.004731
link 23-20 : 0	0
link 23-21 : 0	0
link 23-22 : 0	0
link 23-23 : 0	0
link 23-24 : 0	0
link 23-25 : 0	0
link 23-26 : 0	0
link 23-27 :	0.006913
link 24-0 : 0	0
link 24-1 : 0	0
link 24-2 : 0	0
link 24-3 : 0	0
link 24-4 : 0	0
link 24-5 : 0	0
link 24-6 : 0	0
link 24-7 : 0	0
link 24-8 : 0	0
link 24-9 : 0	0
link 24-10 : 0	0
link 24-11 : 0	0
link 24-12 : 0	0
link 24-13 : 0	0
link 24-14 : 0	0
link 24-15 : 0	0
link 24-16 : 0	0
link 24-17 :	0.011123

link 24-21 : 0	0
link 24-22 : 0	0
link 24-23 : 0	0
link 24-24 : 0	0
link 24-25 : 0	0
link 24-26 :	0.004886
link 24-27 : 0	0
link 25-0 : 0	0
link 25-1 : 0	0
link 25-2 : 0	0
link 25-3 : 0	0
link 25-4 :	0.038289
link 25-5 : 0	0
link 25-6 : 0	0
link 25-7 :	0.005416
link 25-8 : 0	0
link 25-9 : 0	0
link 25-10 : 0	0
link 25-11 : 0	0
link 25-12 : 0	0
link 25-13 : 0	0
link 25-14 : 0	0
link 25-15 : 0	0
link 25-16 : 0	0
link 25-17 : 0	0
link 25-18 : 0	0
link 25-19 : 0	0
link 25-20 : 0	0
link 25-21 : 0	0
link 25-22 :	0.013689
link 25-23 : 0	0
link 25-24 : 0	0
link 25-25 : 0	0
link 25-26 : 0	0
link 25-27 : 0	0
link 26-0 : 0	0
link 26-1 : 0	0
link 26-2 : 0	0

link 24-18 : 0	0
link 24-19 : 0	0
link 24-20 :	0.011108
link 26-6 : 0	0
link 26-7 : 0	0
link 26-8 : 0	0
link 26-9 : 0	0
link 26-10 : 0	0
link 26-11 : 0	0
link 26-12 : 0	0
link 26-13 : 0	0
link 26-14 : 0	0
link 26-15 : 0	0
link 26-16 : 0	0
link 26-17 : 0	0
link 26-18 : 0	0
link 26-19 : 0	0
link 26-20 : 0	0
link 26-21 :	0.002436
link 26-22 : 0	0
link 26-23 : 0	0
link 26-24 :	0.007525
link 26-25 : 0	0
link 26-26 : 0	0
link 26-27 : 0	0
link 27-0 : 0	0
link 27-1 : 0	0
link 27-2 : 0	0
link 27-3 : 0	0
link 27-4 : 0	0
link 27-5 : 0	0
link 27-6 : 0	0
link 27-7 : 0	0
link 27-8 : 0	0
link 27-9 : 0	0
link 27-10 : 0	0
link 27-11 : 0	0
link 27-12 : 0	0
link 27-13 : 0	0
link 27-14 :	0.032313
link 27-15 : 0	0
link 27-16 :	0.032453

link 26-3 :	0.001895
link 26-4 : 0	0
link 26-5 : 0	0

link 27-17 : 0	0
link 27-18 : 0	0
link 27-19 : 0	0
link 27-20 : 0	0
link 27-21 : 0	0
link 27-22 : 0	0
link 27-23 :	0.004862
link 27-24 : 0	0
link 27-25 : 0	0
link 27-26 : 0	0
link 27-27 : 0	0

GERMANY

link 1-11	0	
link 1-12	0	
link 1-13	0	
link 1-14	0	
link 1-15	0	
link 1-16	0	
link 2-0	0	
link 2-1	0	
link 2-2	0	
link 2-3	0	
link 2-4	0.024289	
link 2-5	0	
link 2-6	0	
link 2-7	0	
link 2-8	0	
link 2-9	0.035616	
link 2-10	0	
link 2-11	0	
link 2-12	0	
link 2-13	0.062737	
link 2-14	0	
link 2-15	0	
link 2-16	0	
link 3-0	0	
link 3-1	0	
link 3-2	0	
link 3-3	0	

link 3-16	0	
link 4-0	0	
link 4-1	0	
link 4-2	0.027229	
link 4-3	0.004408	
link 4-4	0	
link 4-5	0	
link 4-6	0	
link 4-7	0	
link 4-8	0	
link 4-9	0	
link 4-10	0	
link 4-11	0	
link 4-12	0	
link 4-13	0	
link 4-14	0	
link 4-15	0	
link 4-16	0	
link 5-0	0	
link 5-1	0	
link 5-2	0	
link 5-3	0	
link 5-4	0	
link 5-5	0	
link 5-6	0	
link 5-7	0	
link 5-8	0	

link 3-4	0.007613	
link 3-5	0	
link 3-6	0	
link 3-7	0	
link 3-8	0	
link 3-9	0.024296	
link 3-10	0	
link 3-11	0	
link 3-12	0	
link 3-13	0	
link 3-14	0	
link 3-15	0	
link 0-0	0	
link 0-1	0	
link 0-2	0	
link 0-3	0	
link 0-4	0	
link 0-5	0	
link 0-6	0.028578	
link 0-7	0.021932	
link 0-8	0	
link 0-9	0	
link 0-10	0.029766	
link 0-11	0	
link 0-12	0	
link 0-13	0	
link 0-14	0	
link 0-15	0	
link 0-16	0	
link 1-0	0	
link 1-1	0	
link 1-2	0	
link 1-3	0	
link 1-4	0	
link 1-5	0	
link 1-6	0.011122	
link 1-7	0.058867	
link 1-8	0	
link 1-9	0	
link 1-10	0	

link 5-9	0.11076	
link 5-10	0.109836	
link 5-11	0.020703	
link 5-12	0	
link 5-13	0	
link 5-14	0.062483	
link 5-15	0	
link 5-16	0	
link 6-0	0.018449	
link 6-1	0.005418	
link 6-2	0	
link 6-3	0	

link 6-4	0	
link 6-5	0	
link 6-6	0	
link 6-7	0.020127	
link 6-8	0	
link 6-9	0	
link 6-10	0	
link 6-11	0	
link 6-12	0	
link 6-13	0	
link 6-14	0	
link 6-15	0	
link 6-16	0	
link 7-0	0.003609	
link 7-1	0.001834	
link 7-2	0	
link 7-3	0	
link 7-4	0	
link 7-5	0.084442	
link 7-6	0.003601	
link 7-7	0	
link 7-8	0	
link 7-9	0.029589	
link 7-10	0.007326	
link 7-11	0	
link 7-12	0	
link 7-13	0	
link 7-14	0	
link 7-15	0	
link 7-16	0	
link 8-0	0	
link 8-1	0	
link 8-2	0	
link 8-3	0	
link 8-4	0	
link 8-5	0	
link 8-6	0	
link 8-7	0	
link 8-8	0	

link 8-9	0	
link 8-10	0	
link 8-11	0.080892	
link 8-12	0	
link 8-13	0	
link 8-14	0	
link 8-15	0.004894	
link 8-16	0	
link 9-0	0	
link 9-1	0	
link 9-2	0.095439	
link 9-3	0.028519	
link 9-4	0	
link 9-5	0.07657	
link 9-6	0	
link 9-7	0	
link 9-8	0	
link 9-9	0	
link 9-10	0	
link 9-11	0	
link 9-12	0	
link 9-13	0	
link 9-14	0	
link 9-15	0	
link 9-16	0	
link 10-0	0.057551	
link 10-1	0	
link 10-2	0	
link 10-3	0	
link 10-4	0	
link 10-5	0.026313	
link 10-6	0	
link 10-7	0.029427	
link 10-8	0	
link 10-9	0	
link 10-10	0	
link 10-11	0	
link 10-12	0	
link 10-13	0	

link 10-14	0.032304	

link 13-2	0	

link 10-15	0	
link 10-16	0	
link 11-0	0	
link 11-1	0	
link 11-2	0	
link 11-3	0	
link 11-4	0	
link 11-5	0.091431	
link 11-6	0	
link 11-7	0	
link 11-8	0.010108	
link 11-9	0	
link 11-10	0	
link 11-11	0	
link 11-12	0	
link 11-13	0	
link 11-14	0	
link 11-15	0	
link 11-16	0	
link 12-0	0	
link 12-1	0	
link 12-2	0	
link 12-3	0	
link 12-4	0	
link 12-5	0.025026	
link 12-6	0	
link 12-7	0	
link 12-8	0	
link 12-9	0	
link 12-10	0	
link 12-11	0	
link 12-12	0	
link 12-13	0	
link 12-14	0.007558	
link 12-15	0	
link 12-16	0.001818	
link 13-0	0	
link 13-1	0.062737	

link 13-3	0	
link 13-4	0	
link 13-5	0	
link 13-6	0	
link 13-7	0	
link 13-8	0	
link 13-9	0	
link 13-10	0	
link 13-11	0	
link 13-12	0	
link 13-13	0	
link 13-14	0	
link 13-15	0	
link 13-16	0	
link 14-0	0	
link 14-1	0	
link 14-2	0	
link 14-3	0	
link 14-4	0	
link 14-5	0	
link 14-6	0	
link 14-7	0	
link 14-8	0	
link 14-9	0	
link 14-10	0	
link 14-11	0	
link 14-12	0.16269	
link 14-13	0	
link 14-14	0	
link 14-15	0.077037	
link 14-16	0	
link 15-0	0	
link 15-1	0	
link 15-2	0	
link 15-3	0	
link 15-4	0	
link 15-5	0	
link 15-6	0	

link 15-7	0	
link 15-8	0.075678	
link 15-9	0	
link 15-10	0	
link 15-11	0	
link 15-12	0	
link 15-13	0	
link 15-14	0.005917	
link 15-15	0	
link 15-16	0.023113	
link 16-0	0	
link 16-1	0	
link 16-2	0	
link 16-3	0	
link 16-4	0	
link 16-5	0	
link 16-6	0	
link 16-7	0	
link 16-8	0	
link 16-9	0	
link 16-10	0	
link 16-11	0	
link 16-12	0.003172	
link 16-13	0	
link 16-14	0	
link 16-15	0.022141	
link 16-16	0	

ROUTING INFORMATION

EUROPE

0,1:1,3,7,20,4,12,0:
0,2:2,14,19,6,0:
0,3:3,7,20,4,12,0:
0,4:4,12,0:
0,5:5,19,6,0:
0,6:6,0:
0,7:7,20,4,12,0:
0,8:8,4,12,0:
0,9:9,11,0:
0,10:10,6,0:
0,11:11,0:
0,12:12,0:
0,13:13,0:
0,14:14,19,6,0:
0,15:15,5,19,6,0:
0,16:16,17,10,6,0:
0,17:17,10,6,0:
0,18:18,8,4,12,0:
0,19:19,6,0:
0,20:20,4,12,0:
0,21:21,16,17,10,6,0:
0,22:22,25,4,12,0:
0,23:23,10,6,0:
0,24:24,17,10,6,0:
0,25:25,4,12,0:
0,26:26,24,17,10,6,0:
0,27:27,14,19,6,0:
1,0:0,6,10,17,16,21,1:
1,2:2,14,27,16,21,1:
1,3:3,1:
1,4:4,17,16,21,1:
1,5:5,19,14,27,16,21,1:
1,6:6,10,17,16,21,1:
1,7:7,3,1:
1,8:8,4,17,16,21,1:
1,9:9,13,19,14,27,16,21,1:
1,10:10,17,16,21,1:
1,11:11,0,6,10,17,16,21,1:
1,12:12,4,17,16,21,1:
1,13:13,19,14,27,16,21,1:
1,14:14,27,16,21,1:
1,15:15,2,14,27,16,21,1:
1,16:16,21,1:
1,17:17,16,21,1:
1,18:18,22,25,7,3,1:
1,19:19,14,27,16,21,1:
1,20:20,7,3,1:
1,21:21,1:
1,22:22,25,7,3,1:
1,23:23,27,16,21,1:
1,24:24,26,3,1:
1,25:25,7,3,1:
1,26:26,3,1:
1,27:27,16,21,1:
2,0:0,6,19,14,2:

2,1:1,21,16,27,14,2:
2,3:3,1,21,16,27,14,2:
2,4:4,17,16,27,14,2:
2,5:5,15,2:
2,6:6,19,14,2:
2,7:7,20,4,17,16,27,14,2:
2,8:8,4,17,16,27,14,2:
2,9:9,13,19,14,2:
2,10:10,6,19,14,2:
2,11:11,0,6,19,14,2:
2,12:12,0,6,19,14,2:
2,13:13,19,14,2:
2,14:14,2:
2,15:15,2:
2,16:16,27,14,2:
2,17:17,16,27,14,2:
2,18:18,8,4,17,16,27,14,2:
2,19:19,14,2:
2,20:20,4,17,16,27,14,2:
2,21:21,16,27,14,2:
2,22:22,25,4,17,16,27,14,2:
2,23:23,19,14,2:
2,24:24,17,16,27,14,2:
2,25:25,4,17,16,27,14,2:
2,26:26,21,16,27,14,2:
2,27:27,14,2:
3,0:0,12,4,20,7,3:
3,1:1,3:

3,2:2,14,27,16,21,1,3:
3,4:4,20,7,3:
3,5:5,19,6,10,17,24,26,3:
3,6:6,10,17,24,26,3:
3,7:7,3:
3,8:8,4,20,7,3:
3,9:9,11,0,12,4,20,7,3:
3,10:10,17,24,26,3:
3,11:11,0,12,4,20,7,3:
3,12:12,4,20,7,3:
3,13:13,0,12,4,20,7,3:
3,14:14,27,16,21,1,3:
3,15:15,2,14,27,16,21,1,3:
3,16:16,21,1,3:
3,17:17,24,26,3:
3,18:18,22,25,7,3:
3,19:19,6,10,17,24,26,3:
3,20:20,7,3:
3,21:21,1,3:
3,22:22,25,7,3:
3,23:23,10,17,24,26,3:
3,24:24,26,3:
3,25:25,7,3:
3,26:26,3:
3,27:27,16,21,1,3:
4,0:0,12,4:
4,1:1,3,7,20,4:
4,2:2,14,19,6,0,12,4:
4,3:3,7,20,4:
4,5:5,19,6,0,12,4:
4,6:6,0,12,4:
4,7:7,20,4:
4,8:8,4:
4,9:9,11,0,12,4:
4,10:10,12,4:
4,11:11,0,12,4:
4,12:12,4:
4,13:13,0,12,4:
4,14:14,19,6,0,12,4:
4,15:15,5,19,6,0,12,4:
4,16:16,17,4:
4,17:17,4:
4,18:18,8,4:
4,19:19,6,0,12,4:
4,20:20,4:
4,21:21,26,24,20,4:
4,22:22,25,4:
4,23:23,10,12,4:
4,24:24,20,4:
4,25:25,4:
4,26:26,24,20,4:
4,27:27,16,17,4:
5,0:0,6,19,5:
5,1:1,21,16,27,14,19,5:
5,2:2,15,5:
5,3:3,26,24,17,10,6,19,5:
5,4:4,12,0,6,19,5:
5,6:6,19,5:
5,7:7,20,4,12,0,6,19,5:
5,8:8,4,12,0,6,19,5:
5,9:9,13,19,5:
5,10:10,6,19,5:
5,11:11,0,6,19,5:

5,12:12,0,6,19,5:
5,13:13,19,5:
5,14:14,19,5:
5,15:15,5:
5,16:16,27,14,19,5:
5,17:17,10,6,19,5:
5,18:18,8,4,12,0,6,19,5:
5,19:19,5:
5,20:20,4,12,0,6,19,5:
5,21:21,16,27,14,19,5:
5,22:22,25,4,12,0,6,19,5:
5,23:23,19,5:
5,24:24,17,10,6,19,5:
5,25:25,4,12,0,6,19,5:
5,26:26,24,17,10,6,19,5:
5,27:27,14,19,5:
6,0:0,6:
6,1:1,3,26,24,17,10,6:
6,2:2,14,19,6:
6,3:3,26,24,17,10,6:
6,4:4,12,0,6:
6,5:5,19,6:
6,7:7,20,4,12,0,6:
6,8:8,4,12,0,6:
6,9:9,11,0,6:
6,10:10,6:
6,11:11,0,6:
6,12:12,0,6:
6,13:13,0,6:
6,14:14,19,6:
6,15:15,5,19,6:
6,16:16,17,10,6:
6,17:17,10,6:
6,18:18,8,4,12,0,6:
6,19:19,6:
6,20:20,4,12,0,6:
6,21:21,16,17,10,6:
6,22:22,25,4,12,0,6:
6,23:23,10,6:
6,24:24,17,10,6:
6,25:25,4,12,0,6:
6,26:26,24,17,10,6:
6,27:27,14,19,6:
7,0:0,12,4,20,7:
7,1:1,3,7:
7,2:2,14,27,16,17,24,20,7:
7,3:3,7:
7,4:4,20,7:
7,5:5,19,6,0,12,4,20,7:
7,6:6,0,12,4,20,7:
7,8:8,4,20,7:
7,9:9,11,0,12,4,20,7:
7,10:10,12,4,20,7:
7,11:11,0,12,4,20,7:
7,12:12,4,20,7:
7,13:13,0,12,4,20,7:
7,14:14,27,16,17,24,20,7:
7,15:15,2,14,27,16,17,24,20,7:
7,16:16,17,24,20,7:
7,17:17,24,20,7:
7,18:18,22,25,7:
7,19:19,6,0,12,4,20,7:
7,20:20,7:
7,21:21,1,3,7:
7,22:22,25,7:
7,23:23,10,12,4,20,7:
7,24:24,20,7:
7,25:25,7:
7,26:26,3,7:
7,27:27,16,17,24,20,7:
8,0:0,12,4,8:
8,1:1,3,7,20,4,8:
8,2:2,14,19,6,0,12,4,8:
8,3:3,7,20,4,8:
8,4:4,8:
8,5:5,19,6,0,12,4,8:
8,6:6,0,12,4,8:
8,7:7,20,4,8:
8,9:9,11,0,12,4,8:
8,10:10,12,4,8:
8,11:11,0,12,4,8:
8,12:12,4,8:
8,13:13,0,12,4,8:
8,14:14,19,6,0,12,4,8:
8,15:15,5,19,6,0,12,4,8:
8,16:16,17,4,8:
8,17:17,4,8:
8,18:18,8:
8,19:19,6,0,12,4,8:
8,20:20,4,8:
8,21:21,26,24,20,4,8:
8,22:22,18,8:

8,23:23,10,12,4,8:
8,24:24,20,4,8:
8,25:25,4,8:
8,26:26,24,20,4,8:
8,27:27,16,17,4,8:
9,0:0,11,9:
9,1:1,3,7,20,4,12,0,11,9:
9,2:2,14,19,13,9:
9,3:3,7,20,4,12,0,11,9:
9,4:4,12,0,11,9:
9,5:5,19,13,9:
9,6:6,0,11,9:
9,7:7,20,4,12,0,11,9:
9,8:8,4,12,0,11,9:
9,10:10,6,0,11,9:
9,11:11,9:
9,12:12,0,11,9:
9,13:13,9:
9,14:14,19,13,9:
9,15:15,5,19,13,9:
9,16:16,17,10,6,0,11,9:
9,17:17,10,6,0,11,9:
9,18:18,8,4,12,0,11,9:
9,19:19,13,9:
9,20:20,4,12,0,11,9:
9,21:21,16,17,10,6,0,11,9:
9,22:22,25,4,12,0,11,9: 9,23:23,19,13,9:
9,24:24,17,10,6,0,11,9:
9,25:25,4,12,0,11,9:
9,26:26,24,17,10,6,0,11,9:
9,27:27,14,19,13,9:
10,0:0,6,10:
10,1:1,3,26,24,17,10:
10,2:2,14,19,6,10:
10,3:3,26,24,17,10:
10,4:4,12,10:
10,5:5,19,6,10:
10,6:6,10:
10,7:7,20,4,12,10:
10,8:8,4,12,10:
10,9:9,11,0,6,10:
10,11:11,0,6,10: 10,12:12,10:
10,13:13,0,6,10:
10,14:14,19,6,10:
10,15:15,5,19,6,10:
10,16:16,17,10:
10,17:17,10:
10,18:18,8,4,12,10:
10,19:19,6,10:
10,20:20,4,12,10:
10,21:21,16,17,10:
10,22:22,25,4,12,10:
10,23:23,10:
10,24:24,17,10:
10,25:25,4,12,10:
10,26:26,24,17,10:
10,27:27,23,10:
11,0:0,11:
11,1:1,3,7,20,4,12,0,11:
11,2:2,14,19,6,0,11:
11,3:3,7,20,4,12,0,11:
11,4:4,12,0,11:
11,5:5,19,6,0,11:
11,6:6,0,11:
11,7:7,20,4,12,0,11:
11,8:8,4,12,0,11:
11,9:9,11:
11,10:10,6,0,11:
11,12:12,0,11:
11,13:13,0,11:
11,14:14,19,6,0,11:
11,15:15,5,19,6,0,11:
11,16:16,17,10,6,0,11:
11,17:17,10,6,0,11:
11,18:18,8,4,12,0,11:
11,19:19,6,0,11:
11,20:20,4,12,0,11:
11,21:21,16,17,10,6,0,11:
11,22:22,25,4,12,0,11:
11,23:23,10,6,0,11:
11,24:24,17,10,6,0,11:
11,25:25,4,12,0,11:
11,26:26,24,17,10,6,0,11:
11,27:27,14,19,6,0,11:
12,0:0,12:
12,1:1,3,7,20,4,12:
12,2:2,14,19,6,0,12:
12,3:3,7,20,4,12: 12,4:4,12:
12,5:5,19,6,0,12:

12,6:6,0,12:
12,7:7,20,4,12: 12,8:8,4,12:
12,9:9,11,0,12:
12,10:10,12:
12,11:11,0,12:
12,13:13,0,12:
12,14:14,19,6,0,12:
12,15:15,5,19,6,0,12:
12,16:16,17,10,12:
12,17:17,10,12:
12,18:18,8,4,12:
12,19:19,6,0,12:
12,20:20,4,12:
12,21:21,16,17,10,12:
12,22:22,25,4,12:
12,23:23,10,12:
12,24:24,17,10,12:
12,25:25,4,12:
12,26:26,24,17,10,12:
12,27:27,23,10,12:
13,0:0,13:
13,1:1,21,16,27,14,19,13:
13,2:2,14,19,13:
13,3:3,7,20,4,12,0,13:
13,4:4,12,0,13:
13,5:5,19,13:
13,6:6,0,13:
13,7:7,20,4,12,0,13:
13,8:8,4,12,0,13:
13,9:9,13:
13,10:10,6,0,13:
13,11:11,0,13:
13,12:12,0,13:
13,14:14,19,13:
13,15:15,5,19,13:
13,16:16,27,14,19,13:
13,17:17,10,6,0,13:
13,18:18,8,4,12,0,13:
13,19:19,13:
13,20:20,4,12,0,13:
13,21:21,16,27,14,19,13:
13,22:22,25,4,12,0,13: 13,23:23,19,13:
13,24:24,17,10,6,0,13:
13,25:25,4,12,0,13:
13,26:26,24,17,10,6,0,13:
13,27:27,14,19,13:
14,0:0,6,19,14:
14,1:1,21,16,27,14:
14,2:2,14:
14,3:3,1,21,16,27,14:
14,4:4,17,16,27,14:
14,5:5,19,14:
14,6:6,19,14:
14,7:7,20,4,17,16,27,14:
14,8:8,4,17,16,27,14:
14,9:9,13,19,14:
14,10:10,6,19,14:
14,11:11,0,6,19,14:
14,12:12,0,6,19,14:
14,13:13,19,14:
14,15:15,2,14:
14,16:16,27,14:
14,17:17,16,27,14:
14,18:18,8,4,17,16,27,14:
14,19:19,14:
14,20:20,4,17,16,27,14:
14,21:21,16,27,14:
14,22:22,25,4,17,16,27,14:
14,23:23,19,14:
14,24:24,17,16,27,14:
14,25:25,4,17,16,27,14:
14,26:26,21,16,27,14:
14,27:27,14:
15,0:0,6,19,5,15:
15,1:1,21,16,27,14,2,15:
15,2:2,15:
15,3:3,1,21,16,27,14,2,15:
15,4:4,12,0,6,19,5,15:
15,5:5,15:
15,6:6,19,5,15:
15,7:7,20,4,12,0,6,19,5,15:
15,8:8,4,12,0,6,19,5,15:
15,9:9,13,19,5,15:
15,10:10,6,19,5,15:
15,11:11,0,6,19,5,15:
15,12:12,0,6,19,5,15:
15,13:13,19,5,15:
15,14:14,2,15:
15,16:16,27,14,2,15:
15,17:17,10,6,19,5,15:

15,18:18,8,4,12,0,6,19,5,15:
15,19:19,5,15:
15,20:20,4,12,0,6,19,5,15:
15,21:21,16,27,14,2,15:
15,22:22,25,4,12,0,6,19,5,15:
15,23:23,19,5,15:
15,24:24,17,10,6,19,5,15:
15,25:25,4,12,0,6,19,5,15:
15,26:26,21,16,27,14,2,15:
15,27:27,14,2,15:
16,0:0,6,10,17,16:
16,1:1,21,16:
16,2:2,14,27,16:
16,3:3,1,21,16:
16,4:4,17,16:
16,5:5,19,14,27,16:
16,6:6,10,17,16:
16,7:7,20,4,17,16:
16,8:8,4,17,16:
16,9:9,13,19,14,27,16:
16,10:10,17,16:
16,11:11,0,6,10,17,16:
16,12:12,4,17,16:
16,13:13,19,14,27,16:
16,14:14,27,16:
16,15:15,2,14,27,16:
16,17:17,16:
16,18:18,8,4,17,16:
16,19:19,14,27,16:
16,20:20,4,17,16:
16,21:21,16:
16,22:22,25,4,17,16:
16,23:23,27,16:
16,24:24,17,16:
16,25:25,4,17,16:
16,26:26,21,16:
16,27:27,16:
17,0:0,6,10,17:
17,1:1,3,26,24,17:
17,2:2,14,27,16,17:
17,3:3,26,24,17:
17,4:4,17:
17,5:5,19,6,10,17:
17,6:6,10,17:
17,7:7,20,4,17:
17,8:8,4,17:
17,9:9,11,0,6,10,17:
17,10:10,17:
17,11:11,0,6,10,17:
17,12:12,4,17:
17,13:13,0,6,10,17:
17,14:14,27,16,17:
17,15:15,2,14,27,16,17:
17,16:16,17:
17,18:18,8,4,17:
17,19:19,6,10,17:
17,20:20,4,17:
17,21:21,16,17:
17,22:22,25,4,17:
17,23:23,10,17:
17,24:24,17:
17,25:25,4,17:
17,26:26,24,17:
17,27:27,16,17:
18,0:0,12,4,8,18:
18,1:1,3,7,25,22,18:
18,2:2,14,19,6,0,12,4,8,18:
18,3:3,7,25,22,18:
18,4:4,8,18:
18,5:5,19,6,0,12,4,8,18:
18,6:6,0,12,4,8,18:
18,7:7,25,22,18:
18,8:8,18:
18,9:9,11,0,12,4,8,18:
18,10:10,12,4,8,18:
18,11:11,0,12,4,8,18:
18,12:12,4,8,18:
18,13:13,0,12,4,8,18:
18,14:14,19,6,0,12,4,8,18:
18,15:15,5,19,6,0,12,4,8,18:
18,16:16,17,4,8,18:
18,17:17,4,8,18:
18,19:19,6,0,12,4,8,18:
18,20:20,4,8,18:
18,21:21,1,3,7,25,22,18:
18,22:22,18:
18,23:23,10,12,4,8,18:
18,24:24,20,4,8,18:
18,25:25,22,18:
18,26:26,3,7,25,22,18:
18,27:27,16,17,4,8,18:
19,0:0,6,19:

19,1:1,21,16,27,14,19:
19,2:2,14,19:
19,3:3,26,24,17,10,6,19:
19,4:4,12,0,6,19:
19,5:5,19:
19,6:6,19:
19,7:7,20,4,12,0,6,19:
19,8:8,4,12,0,6,19:
19,9:9,13,19:
19,10:10,6,19:
19,11:11,0,6,19:
19,12:12,0,6,19:
19,13:13,19:
19,14:14,19:
19,15:15,5,19:
19,16:16,27,14,19:
19,17:17,10,6,19:
19,18:18,8,4,12,0,6,19:
19,20:20,4,12,0,6,19:
19,21:21,16,27,14,19:
19,22:22,25,4,12,0,6,19:
19,23:23,19:
19,24:24,17,10,6,19:
19,25:25,4,12,0,6,19:
19,26:26,24,17,10,6,19:
19,27:27,14,19:
20,0:0,12,4,20:
20,1:1,3,7,20:
20,2:2,14,27,16,17,24,20:
20,3:3,7,20:
20,4:4,20:
20,5:5,19,6,0,12,4,20:
20,6:6,0,12,4,20:
20,7:7,20:
20,8:8,4,20:
20,9:9,11,0,12,4,20:
20,10:10,12,4,20:
20,11:11,0,12,4,20: 20,12:12,4,20:
20,13:13,0,12,4,20:
20,14:14,27,16,17,24,20:
20,15:15,2,14,27,16,17,24,20:
20,16:16,17,24,20:
20,17:17,24,20:
20,18:18,8,4,20:
20,19:19,6,0,12,4,20:
20,21:21,26,24,20:
20,22:22,25,4,20:
20,23:23,10,12,4,20:
20,24:24,20:
20,25:25,4,20:
20,26:26,24,20:
20,27:27,16,17,24,20:
21,0:0,6,10,17,16,21:
21,1:1,21:
21,2:2,14,27,16,21:
21,3:3,1,21:
21,4:4,17,16,21:
21,5:5,19,14,27,16,21:
21,6:6,10,17,16,21:
21,7:7,3,1,21:
21,8:8,4,17,16,21:
21,9:9,13,19,14,27,16,21:
21,10:10,17,16,21:
21,11:11,0,6,10,17,16,21:
21,12:12,4,17,16,21:
21,13:13,19,14,27,16,21:
21,14:14,27,16,21:
21,15:15,2,14,27,16,21:
21,16:16,21:
21,17:17,16,21:
21,18:18,8,4,17,16,21:
21,19:19,14,27,16,21:
21,20:20,24,26,21:
21,22:22,25,7,3,1,21:
21,23:23,27,16,21:
21,24:24,26,21:
21,25:25,7,3,1,21:
21,26:26,21:
21,27:27,16,21:
22,0:0,12,4,25,22:
22,1:1,3,7,25,22:
22,2:2,14,19,6,0,12,4,25,22:
22,3:3,7,25,22:
22,4:4,25,22:
22,5:5,19,6,0,12,4,25,22:
22,6:6,0,12,4,25,22:
22,7:7,25,22:
22,8:8,18,22:
22,9:9,11,0,12,4,25,22:
22,10:10,12,4,25,22:
22,11:11,0,12,4,25,22:

22,12:12,4,25,22:
22,13:13,0,12,4,25,22:
22,14:14,19,6,0,12,4,25,22:
22,15:15,5,19,6,0,12,4,25,22:
22,16:16,17,4,25,22:
22,17:17,4,25,22:
22,18:18,22:
22,19:19,6,0,12,4,25,22:
22,20:20,4,25,22:
22,21:21,1,3,7,25,22:
22,23:23,10,12,4,25,22:
22,24:24,20,4,25,22:
22,25:25,22:
22,26:26,3,7,25,22:
22,27:27,16,17,4,25,22:
23,0:0,6,10,23:
23,1:1,21,16,27,23:
23,2:2,14,19,23:
23,3:3,26,24,17,10,23:
23,4:4,12,10,23:
23,5:5,19,23:
23,6:6,10,23:
23,7:7,20,4,12,10,23:
23,8:8,4,12,10,23:
23,9:9,13,19,23:
23,10:10,23:
23,11:11,0,6,10,23:
23,12:12,10,23:
23,13:13,19,23:
23,14:14,19,23:
23,15:15,5,19,23:
23,16:16,27,23:
23,17:17,10,23:
23,18:18,8,4,12,10,23:
23,19:19,23:
23,20:20,4,12,10,23:
23,21:21,16,27,23:
23,22:22,25,4,12,10,23:
23,24:24,17,10,23:
23,25:25,4,12,10,23:
23,26:26,24,17,10,23:
23,27:27,23:
24,0:0,6,10,17,24:
24,1:1,3,26,24:
24,2:2,14,27,16,17,24:
24,3:3,26,24:
24,4:4,17,24:
24,5:5,19,6,10,17,24:
24,6:6,10,17,24:
24,7:7,20,24:
24,8:8,4,17,24:
24,9:9,11,0,6,10,17,24:
24,10:10,17,24:
24,11:11,0,6,10,17,24:
24,12:12,4,17,24:
24,13:13,0,6,10,17,24:
24,14:14,27,16,17,24:
24,15:15,2,14,27,16,17,24:
24,16:16,17,24:
24,17:17,24:
24,18:18,8,4,17,24:
24,19:19,6,10,17,24:
24,20:20,24:
24,21:21,26,24:
24,22:22,25,4,17,24:
24,23:23,10,17,24:
24,25:25,4,17,24:
24,26:26,24:
24,27:27,16,17,24:
25,0:0,12,4,25:
25,1:1,3,7,25:
25,2:2,14,19,6,0,12,4,25:
25,3:3,7,25:
25,4:4,25:
25,5:5,19,6,0,12,4,25:
25,6:6,0,12,4,25:
25,7:7,25:
25,8:8,4,25:
25,9:9,11,0,12,4,25:
25,10:10,12,4,25:
25,11:11,0,12,4,25:
25,12:12,4,25:
25,13:13,0,12,4,25:
25,14:14,19,6,0,12,4,25:
25,15:15,5,19,6,0,12,4,25:
25,16:16,17,4,25:
25,17:17,4,25:
25,18:18,22,25:
25,19:19,6,0,12,4,25:
25,20:20,4,25:
25,21:21,1,3,7,25:
25,22:22,25:

25,23:23,10,12,4,25:
25,24:24,20,4,25:
25,26:26,3,7,25:
25,27:27,16,17,4,25:
26,0:0,6,10,17,24,26:
26,1:1,3,26:
26,2:2,14,27,16,21,26:
26,3:3,26:
26,4:4,17,24,26:
26,5:5,19,6,10,17,24,26:
26,6:6,10,17,24,26:
26,7:7,3,26:
26,8:8,4,17,24,26:
26,9:9,11,0,6,10,17,24,26:
26,10:10,17,24,26:
26,11:11,0,6,10,17,24,26: 26,12:12,4,17,24,26:
26,13:13,0,6,10,17,24,26:
26,14:14,27,16,21,26:
26,15:15,2,14,27,16,21,26:
26,16:16,21,26:
26,17:17,24,26:
26,18:18,22,25,7,3,26:
26,19:19,6,10,17,24,26:
26,20:20,24,26:
26,21:21,26:
26,22:22,25,7,3,26:
26,23:23,10,17,24,26:
26,24:24,26:
26,25:25,7,3,26:
26,27:27,16,21,26:
27,0:0,6,10,23,27:
27,1:1,21,16,27:
27,2:2,14,27:
27,3:3,1,21,16,27:
27,4:4,17,16,27:
27,5:5,19,14,27:
27,6:6,10,23,27:
27,7:7,20,4,17,16,27:
27,8:8,4,17,16,27:
27,9:9,13,19,14,27:
27,10:10,23,27:
27,11:11,0,6,10,23,27:
27,12:12,10,23,27:
27,13:13,19,14,27:
27,14:14,27:
27,15:15,2,14,27:
27,16:16,27:
27,17:17,16,27:
27,18:18,8,4,17,16,27:
27,19:19,14,27:
27,20:20,4,17,16,27:
27,21:21,16,27:
27,22:22,25,4,17,16,27:
27,23:23,27:
27,24:24,17,16,27:
27,25:25,4,17,16,27:
27,26:26,21,16,27:

GERMANY

0,1:1,6,0:
0,2:2,9,5,10,0:
0,3:3,9,5,10,0:
0,4:4,2,9,5,10,0:
0,5:5,10,0:
0,6:6,0:
0,7:7,0:
0,8:8,11,5,10,0:
0,9:9,5,10,0:
0,10:10,0:
0,11:11,5,10,0:
0,12:12,5,10,0:
0,13:13,1,6,0:
0,14:14,15,8,11,5,10,0:
0,15:15,8,11,5,10,0:
0,16:16,15,8,11,5,10,0:
1,0:0,6,1:
1,2:2,13,1:
1,3:3,4,2,13,1:
1,4:4,2,13,1:
1,5:5,9,2,13,1:
1,6:6,1:
1,7:7,1:
1,8:8,11,5,9,2,13,1:
1,9:9,2,13,1:
1,10:10,0,6,1:
1,11:11,5,9,2,13,1:
1,12:12,5,9,2,13,1:

1,13:13,1:
1,14:14,15,8,11,5,9,2,13,1: 1,15:15,8,11,5,9,2,13,1:
1,16:16,15,8,11,5,9,2,13,1:
2,0:0,7,9,2:
2,1:1,7,9,2:
2,3:3,4,2: 2,4:4,2:
2,5:5,9,2:
2,6:6,7,9,2:
2,7:7,9,2:
2,8:8,11,5,9,2:
2,9:9,2:
2,10:10,5,9,2: 2,11:11,5,9,2:
2,12:12,5,9,2:
2,13:13,2:
2,14:14,15,8,11,5,9,2: 2,15:15,8,11,5,9,2:
2,16:16,15,8,11,5,9,2:
3,0:0,7,9,3:
3,1:1,7,9,3:
3,2:2,4,3: 3,4:4,3:
3,5:5,9,3:
3,6:6,7,9,3:
3,7:7,9,3:
3,8:8,11,5,9,3:
3,9:9,3:
3,10:10,5,9,3: 3,11:11,5,9,3:
3,12:12,5,9,3:
3,13:13,2,4,3:
3,14:14,15,8,11,5,9,3: 3,15:15,8,11,5,9,3:
3,16:16,15,8,11,5,9,3:
4,0:0,7,9,2,4:
4,1:1,7,9,2,4:
4,2:2,4:
4,3:3,4:
4,5:5,9,2,4:
4,6:6,7,9,2,4:
4,7:7,9,2,4:
4,8:8,11,5,9,2,4:
4,9:9,2,4:
4,10:10,5,9,2,4:
4,11:11,5,9,2,4:
4,12:12,5,9,2,4:
4,13:13,2,4:
4,14:14,15,8,11,5,9,2,4:
4,15:15,8,11,5,9,2,4:
4,16:16,15,8,11,5,9,2,4:
5,0:0,7,5:
5,1:1,7,5:
5,2:2,9,5:
5,3:3,9,5:
5,4:4,2,9,5:
5,6:6,7,5:
5,7:7,5:
5,8:8,11,5:
5,9:9,5:
5,10:10,5: 5,11:11,5:
5,12:12,5:
5,13:13,1,7,5:
5,14:14,15,8,11,5:
5,15:15,8,11,5:
5,16:16,15,8,11,5:
6,0:0,6:
6,1:1,6:
6,2:2,13,1,6:
6,3:3,4,2,13,1,6:
6,4:4,2,13,1,6:
6,5:5,10,0,6:
6,7:7,6:
6,8:8,11,5,10,0,6:
6,9:9,2,13,1,6:
6,10:10,0,6:
6,11:11,5,10,0,6:
6,12:12,5,10,0,6:
6,13:13,1,6:
6,14:14,15,8,11,5,10,0,6:
6,15:15,8,11,5,10,0,6:
6,16:16,15,8,11,5,10,0,6:
7,0:0,7:
7,1:1,7:
7,2:2,13,1,7:
7,3:3,9,5,10,7:
7,4:4,2,13,1,7:
7,5:5,10,7:
7,6:6,7:
7,8:8,11,5,10,7:

7,9:9,5,10,7:
7,10:10,7:
7,11:11,5,10,7:
7,12:12,5,10,7:
7,13:13,1,7:
7,14:14,15,8,11,5,10,7: 7,15:15,8,11,5,10,7:
7,16:16,15,8,11,5,10,7:
8,0:0,7,5,11,8:
8,1:1,7,5,11,8:
8,2:2,9,5,11,8:
8,3:3,9,5,11,8:
8,4:4,2,9,5,11,8:
8,5:5,11,8:
8,6:6,7,5,11,8:
8,7:7,5,11,8:
8,9:9,5,11,8:
8,10:10,5,11,8:
8,11:11,8:
8,12:12,14,15,8:
8,13:13,1,7,5,11,8:
8,14:14,15,8: 8,15:15,8:
8,16:16,15,8:
9,0:0,7,9:
9,1:1,7,9:
9,2:2,9:
9,3:3,9:
9,4:4,2,9: 9,5:5,9:
9,6:6,7,9:
9,7:7,9:
9,8:8,11,5,9:
9,10:10,5,9: 9,11:11,5,9:
9,12:12,5,9:
9,13:13,2,9:
9,14:14,15,8,11,5,9: 9,15:15,8,11,5,9:
9,16:16,15,8,11,5,9:
10,0:0,10:
10,1:1,7,10:
10,2:2,9,5,10:
10,3:3,9,5,10:
10,4:4,2,9,5,10:
10,5:5,10:
10,6:6,0,10:
10,7:7,10:
10,8:8,11,5,10:
10,9:9,5,10:
10,11:11,5,10:
10,12:12,5,10:
10,13:13,1,7,10:
10,14:14,15,8,11,5,10:
10,15:15,8,11,5,10:
10,16:16,15,8,11,5,10:
11,0:0,7,5,11:
11,1:1,7,5,11:
11,2:2,9,5,11:
11,3:3,9,5,11:
11,4:4,2,9,5,11:
11,5:5,11:
11,6:6,7,5,11:
11,7:7,5,11: 11,8:8,11:
11,9:9,5,11:
11,10:10,5,11:
11,12:12,14,15,8,11:
11,13:13,1,7,5,11:
11,14:14,15,8,11:
11,15:15,8,11:
11,16:16,15,8,11:
12,0:0,10,14,12:
12,1:1,7,5,14,12:
12,2:2,9,5,14,12:
12,3:3,9,5,14,12:
12,4:4,2,9,5,14,12:
12,5:5,14,12:
12,6:6,0,10,14,12:
12,7:7,5,14,12:
12,8:8,15,14,12:
12,9:9,5,14,12:
12,10:10,14,12:
12,11:11,5,14,12:
12,13:13,1,7,5,14,12:
12,14:14,12:
12,15:15,14,12:
12,16:16,12:
13,0:0,7,9,2,13:
13,1:1,7,9,2,13:
13,2:2,13:
13,3:3,4,2,13:

13,4:4,2,13:
13,5:5,9,2,13:
13,6:6,7,9,2,13:
13,7:7,9,2,13:
13,8:8,11,5,9,2,13:
13,9:9,2,13:
13,10:10,5,9,2,13: 13,11:11,5,9,2,13:
13,12:12,5,9,2,13:
13,14:14,15,8,11,5,9,2,13: 13,15:15,8,11,5,9,2,13:
13,16:16,15,8,11,5,9,2,13:
14,0:0,10,14:
14,1:1,7,5,14:
14,2:2,9,5,14:
14,3:3,9,5,14:
14,4:4,2,9,5,14:
14,5:5,14:
14,6:6,0,10,14:
14,7:7,5,14:
14,8:8,15,14:
14,9:9,5,14:
14,10:10,14:
14,11:11,5,14:
14,12:12,14:
14,13:13,1,7,5,14:
14,15:15,14:
14,16:16,12,14:
15,0:0,10,14,15:
15,1:1,7,5,14,15:
15,2:2,9,5,14,15:
15,3:3,9,5,14,15:
15,4:4,2,9,5,14,15:
15,5:5,14,15:
15,6:6,0,10,14,15:
15,7:7,5,14,15: 15,8:8,15:
15,9:9,5,14,15:
15,10:10,14,15: 15,11:11,8,15:
15,12:12,14,15:
15,13:13,1,7,5,14,15:
15,14:14,15:
15,16:16,15:
16,0:0,10,14,15,16:
16,1:1,7,5,14,15,16:
16,2:2,9,5,14,15,16:
16,3:3,9,5,14,15,16:
16,4:4,2,9,5,14,15,16:
16,5:5,14,15,16:
16,6:6,0,10,14,15,16:
16,7:7,5,14,15,16:
16,8:8,15,16:
16,9:9,5,14,15,16:
16,10:10,14,15,16:
16,11:11,8,15,16:
16,12:12,16:
16,13:13,1,7,5,14,15,16:
16,14:14,15,16: