OPTIMAL TOPOLOGY DESIGN FOR VIRTUAL NETWORKS

by

Mina Nabil Youssef

B.S., Alexandria University, Alexandria, Egypt, 2004

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Electrical and Computer Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2008

Approved by:

Major Professor
Caterina Scoglio

Abstract

Recently, virtualization was proposed in many scientific fields. Virtualization is widely applied in telecommunications where networks are required to be extremely flexible to meet the current and the unpredictable future requirements. The creation of a virtual network over the physical network allows the application developers to design new services provided to the users without modifying the underlay resources. The creation of a virtual network of light paths and light trees over the optical network allows the resources managers to utilize the huge optical capacity more efficiently.

In this thesis, we consider the optimal topology design for the virtual networks taking into consideration traffic demands and quality of service constraints of the applications. Considered examples of virtual networks are the overlay networks at the application layer and the virtual light path and light tree networks at the optical layer.

In the design of overlay topologies, the performance of the virtual networks is affected by traffic characteristic, and behavior of nodes which can be selfish or cooperative. Both the static and dynamic traffic demand scenarios are considered. The static demand scenario follows well known probability distributions, while in the dynamic traffic scenario, the traffic matrix is predicted through measurements over each link in the network. We study the problem of finding the overlay topology that minimizes a cost function which takes into account the overlay link creation cost and the routing cost. We formulate the problem as an Integer Linear Programming and propose heuristics to find near-optimal overlay topologies with a reduced complexity.

Virtual optical networks are designed to support many applications. Multicast sessions are an example of the applications running over the optical network. The main objective in creating the hybrid topology, composed by light paths and light trees, is to increase number of supported multicast sessions through sharing the network resources. The problem of establishing the hybrid topology is formulated using the Integer Linear Programming. Extensive data results and analysis are performed on the generated hybrid topologies for evaluation.

Table of Contents

List of Figures

List of Tables

Acknowledgments

We want to thank everybody who helps and supports us to come up with this thesis. First of all, we indeed thank Dr. Caterina Scoglio for her guidance, effort, time and help during the master program to make this research work. We also thank Dr. Todd Easton, Dr. Don Gruenbacher and Dr. Bala Natarajan for their guidance and comments on this work. We also thank Dr. Baek-Young Choi for her help, guidace and effort. For their help, we would like to thank the former and current postgraduate students in the Sunflower Networking Group for their support and comments and especially we would like to thank Ben McBride for his help and efforts. We can not forget to thank all the visitor professors who gave us their opinions about the research work. For their support we would like to thank Dr. Jennifer Rexford and Dr. Tricha Anjali. Finally, we can not forget Dr. Medhat Morcos for his help and support.

Dedication

To

JESUS CHRIST who is my LORD and my SAVIOR.

Saint Mary the Theotokos.

Saint Mark the Evangelist, the Martyr and the first Pope of Alexandria.

Saint George the prince of Martyrs.

Saint Mina the Martyr the Wonder-worker.

Saint Cosmas and Saint Damian the physicians.

Saint Antony the father of monks.

Saint Hermina the Anchorite.

Pope Cyril VI of Alexandria.

Pope Shenouda III of Alexandria.

To my parents Nabil and Angel, my sisters Miral and Mariam.

To my uncle Father Andrew Khalil and my aunt Nadia.

Chapter 1

Introduction

1.1 Optimal Overlay Topology Design

1.1.1 Motivation

Peer-to-peer and many multimedia applications have recently grown with the need for high Quality of Service (QoS) [1], [2], [3], [4], [5], [6]. Providing the required quality of service for these applications over a packet switching network has been a critical task for a long time.

A recent approach for providing QoS without changing the network architecture is based on the use of overlay networks. An overlay network is an application-layer logical network created on top of the physical network. It is formed by all or a subset of the underlying physical nodes. The connections between each pair of overlay nodes are provided by overlay links which consist of many underlying physical links. Overlay networks can be used to improve performance and provide quality of service on the Internet Protocol network, by routing data on the overlay links based on performance measurements. Among the most interesting open problems in overlay network design is the topology creation such as node location and link setup.

1.1.2 Contribution

We consider the problem of finding the overlay topology that minimizes a cost function, which is given by the weighted sum of the overlay link creation cost and the routing cost. The routing cost is assumed to be proportional to the traffic demand.

First, we formulate the problem as an Integer Linear Programming (ILP) for a given traffic matrix in case of cooperative (C node) and non-cooperative (N-C node) behavior. We assume that the nodes act non cooperatively, so each node establishes overlay links to send only its traffic demands. The N-C node behavior is assumed to avoid the phenomenon of the *free riding*. Following [7], it has been noticed that in overlay topologies, few nodes establish most of the links and all the other nodes use those links to route their traffic. Consequently the resulting topology has few nodes with high degree, leading to a non-robust and unbalanced topology. The assumption of non-cooperative node behavior avoids transit traffic to be routed on newly created overlay links. On the other hand, if we consider that each node establishes overlay links to send its traffic demands and to allow other nodes to route their traffic demands over them, the nodes act cooperatively. Both behaviors are considered when minimizing the overall network cost. The solutions of the ILP problem in average-size networks are analyzed, showing that the amount of traffic demands between the nodes affect the decision of creating new overlay links, and the resulting optimal topologies are different from the regular topologies obtained when neglecting traffic demands.
Furthermore, some heuristics are proposed to find near-optimal overlay topologies with a reduced complexity. Some heuristics are based on the selection of the best destination toward which to build an overlay link. Some heuristics are based on traffic volume, number of hops and a combination of both. Another heuristic is based on clustering the nodes and assigning leaders for each cluster. An additional heuristic allows each node to create new overlay links, where nodes are considered in a certain sequence. Additionally, a heuristic is based on comparing between the overlay link creation cost and the transport cost. Finally, a heuristic is based on Dijkstra algorithm given a fully mesh overlay topology with weighted links. The link weight is a function of the traffic demand, overlay cost coefficient and the shortest path on the overlay network.

Extensive testing and simulations are done on the heuristics to compare the generated topology with the optimal ones given different traffic demands scenarios. Guidelines for the selection of the best heuristic among the set of the proposed ones, as a function of the cost weight, are also provided.

Summarizing, the contributions in this thesis are:

1. Formulating the problem of establishing new overlay links in the network using ILP.

2. Proposing some heuristics to generate near optimal overlay topology.

3. Characterizing the generated optimal and near optimal topologies.

1.2 Adaptation of Overlay Network Topology

1.2.1 Motivation

Adapting the overlay topology based on the flows between origin-destination (OD) node pairs in the overlay network is an important problem in overlay network design. We believe that estimating and predicting the OD flows can enhance the performance of the overlay network by reconfiguring the topology to minimize the transportation cost and the overlay link creation cost. Estimating the traffic matrix can enhance the management of a network by identifying link failures and tracking traffic loads for capacity planning purposes. We consider traffic matrix estimation and prediction as a new approach to design dynamic overlay networks based on the change in the traffic volume passing over the underlay networks.

1.2.2 Contribution

We study the problem of adapting the overlay network topology based on the estimation and the prediction of traffic volume in the network. Estimating the OD pairs flows allow the design of dynamic overlay network topology according to the change in the traffic demands. Kalman filter is used as a prediction tool to track the change in the traffic demand.

Predicting the traffic flows of the OD pair flows help the network operator to decide about creating or dropping overlay links between the OD pairs to minimize the cost function. The cost function is composed of the cost of creating and maintaining the overlay links and the cost of flow transport. Based on the estimated traffic from the observed data, we can keep the overlay network cost at low levels all the time by dynamically changing the topology. In contrast to the work in [8], we will not consider detecting the abnormal traffic demands because the effect of the abnormal traffic demands on the overlay network topology is avoided by the cooperativeness behavior of nodes when creating the overlay topology. The overlay topology design depends on the overlay cost coefficient α and the traffic demands [9] according to which links are created to minimize the global cost function. Given the value of α and the prediction of the traffic volumes, the overlay network topology is established to minimize its cost function. We propose a heuristic to create the overlay topology considering the co operative behavior of the nodes in the network. The heuristic results show that the cost function of both the optimal overlay topology and the near optimal overlay topology are so closed.

1.3 Hybrid Optical Topology Design for Supporting Multicast Sessions

1.3.1 Motivation

Management of network resources has been important issue since the growth of the traffic demand became exponentially. With the increasing of the traffic demands, the optical networks are upgraded with tremendous bandwidth. Multicast traffic demand is one type of the most dominant traffic running on the optical network. IP layer multicast in the IP-networks supports multicast session demands without exploring the capability of the underlay networks. Wavelength Division Multiplexing networks are able to support multicast sessions with efficient use of the optical network resources. In the IP multicasting, multicast session packets are copied to different outgoing links at the routers electronically. The conversion between optical signal to electronic signal to optical signal (O/E/O) introduces latency which contributes in the total delay in the network.

Different techniques were proposed to support the IP multicast over the WDM optical network. Light path and light tree were proposed to efficiently use the optical channels over the optical networks. A light path is established from an electronic-to-optic node (E/O) and it ends at an optic-to-electronic node (O/E). The light path engages an optical channel over each optical link over which the light path is established. The light path mainly decreases number of O/E/O conversion. The problem of the light path is the scalability in case of a large number of multicast memebers.

A light tree is a tree established over the WDM layer. The light tree is a general structure of the light path where branching at the optical devices is allowed with the support of wavelength converters and optical signal splitters. A light tree is composed by a group of optical nodes such that they receive the same packets. Like the light path, the light tree engages an optical channel over all optical links composing it. Multicast sessions use the light trees taking into consideration that the non destination nodes in the light tree will receive an unecessary copy of the multicast packets.

1.3.2 Contribution

When a light path or a light tree is used to support a multicast session, it uses the wavelength channel allocated to it entirely. The concept of sharing the wavelength channel over the physical links and light paths was proposed in [10] where each wavelength channel is divided to sub-wavelength to support more multicast sessions. Sharing the wavelength channels between different multicast sessions increases the flexibility in maximizing the utilization of the channel resources in the WDM optical network.

In this work, a hybrid topology design over the WDM networks is proposed to support multicast sessions. The hybrid topology is composed of light paths, light trees and the physical links. For each light tree and light path, a degree of sharing the wavelength channel is defined. The implementation of the light trees and the light on the optical links allows us to find the available wavelength channels on each optical link. Light paths and light trees are implemented based on the shortest path between the members. The light trees could represent tree topologies of previous multicast sessions and those sessions are no longer active, or they could be assumed in the problem given light tree members and the shortest path trees between the light tree members represent the light tree.

The problem of creating hybrid topology over WDM optical network is formulated as a Mixed Integer Linear Programming MILP to support new multicast sessions. The problem is formulated on two steps: 1)computing the available wavelength channels given a set of light trees and light paths (in general, we can say light trees because light path is a special case of light tree) and the total number of wavelength channels over the physical links, 2) given the available wavelength channels on the physical links, light paths, light trees and the degree of sharing the wavelength on the light paths and light trees, we construct the hybrid multicast session topology. Some heuristics are proposed to support the multicast sessions given the physical and logical links.

Extensive testing and simulations are done on the heuristics to compare the generated topology with the optimal ones given different multicast sessions. our contribution for that part is formulating the problem of establishing hybrid topology over the WDM optical network to support multicast sesions.

The thesis is organized by discussing the optimal topology design of overlay network in chapter 2. In chapter 3, we discuss the adaptation of the overlay topology based on the change in the traffic matrix. Design of hybrid optical network topology for supporting multicast is discussed in chapter 4. Finally, we conclude and give some guidelines for the future work in chapter 5.

Chapter 2

Optimal Overlay Topology Design

2.1 Introduction

Many internet applications are in need of support from the underlay networks. While the underlay network is growing, some applications can not get the desirable support and can not offer the request services to the end users. The overlay network is proposed to support applications, which are in need for high quality of services. An overlay network is an application-layer logical network created on top of the physical network. It is formed by all or a subset of the underlying physical nodes. The connections between each pair of overlay nodes are provided by overlay links, which consist of many underlying physical links. Both the overlay nodes and the overlay links compose the overlay topology. In this chapter, we create optimal and near optimal overlay network topologies given a cost function. The cost function reflect the functionality of creating overlay links and routing the traffic demand over the underlay link. Traffic demands between peers are assumed in the network following different traffic scenarios.

The chapter flow starts with discussing the related work in section 2.2. In section 2.3, we define the cost function and the ILP formulation of the optimal overlay network topology. In section 2.4, we present the proposed heuristics. The underlay networks characteristics, some topological characteristics and the traffic demand model are thoroughly described in section 2.5. In section 2.6, we show and explain the results of both the ILP problem formulation and the proposed heuristics.

2.2 Related Work

In designing the overlay topology [7], node behavior can be considered selfish. In the selfish behavior, nodes establish links in order to minimize their own costs. Consequently the global overlay network obtained by selfish nodes can be different from the optimal global network that could be created if the nodes behave in a cooperative way. This difference is called the *cost of Anarchy*. Selfish and non-selfish behaviors of the nodes in the networks have a great impact on the selection of the topology and its cost.

The cost function used in [7] does not consider the demand volume between nodes as an important factor. Instead, we believe that when considering traffic demands, it is possible to obtain topologies that have better characteristics with respect to some keys graph-theoretic metrics introduced in [11], such as node degree distributions, diameter and clustering coefficient.

In [12], the authors consider the static and the dynamic overlay topology design problems. The static overlay topology design is applied when there are no changes in the traffic requirements. In case that the communication requirements change over the time, the authors consider the dynamic overlay topology design based on two cost components: occupancy cost and reconfiguration cost. However this approach is suited for service overlay networks, where an overlay service provider designs the overlay network.

The authors in [13] address many topics concerning selfish routing in Internet-like environments. They use the fully connected overlay topology to limit the parameter space and to reduce the complexity of the problem. They study the performance of the selfish overlay routing when all the network nodes are included in the overlay network. Routing constraints are shown to have little effect on the network-wide cost when varying network load.

In [14], the authors show the effect of the traffic demands on the overlay topology given different scenarios of the traffic demands. They consider a fully connected underlay topology over which the overlay topology was built. Some of the obtained overlay topologies are not resilient to targeted node failures or attacks.

The construction of resilient service overlay network (SON) under path failure in the physical network and under performance degradation has been studied in [15]. The authors concluded that the performance of overlay routing service highly depends on the construction of overlay topologies. They did not mention the effect of the traffic demands on the performance of overlay routing service.

In our previous work [9], we studied the creation of optimal overlay topologies taking into account homogeneous and randomly uniform traffic demands in the network. The resulting optimal overlay topologies are different from the regular topologies obtained when neglecting traffic demands among nodes.

The goal of this chapter is to study the problem of optimal overlay topology design taking into account traffic demands, and to analyze the characteristics of the obtained optimal and near optimal overlay topologies in order to provide simple guidelines for the overlay topology design.

2.3 Overlay Topology Design

2.3.1 Problem Formulation

Overlay networks are created at the application layer, over a given physical network. Overlay network nodes select their neighbors and establish direct overlay links creating an overlay topology. Let $G_u = (N, E)$ be the graph representing the underlay, or physical network and $G = (N, L)$ be the graph representing the overlay network. We have assumed that the same set of nodes N are in both the overlay and physical networks, while the set of overlay links can be different from the set of physical links E. We define the default topology as the overlay topology having $L \equiv E$ where all underlay links are also overlay links. Any logical link in L is setup on a path $l_{i,j}$ composed by physical links on the shortest paths between node i and node j. Assuming that each node $i \in N$ has a traffic demand toward a node subset $S_i \subset N$, let $d_{i,j}$ be the traffic demand between node i and node j in the subset S_i. The objective of each node is to create logical links to be connected with all nodes in S_i such that the total cost is minimized.

The total cost function is composed of two components:

1. The cost to create an overlay link between a pair of node is proportional to the number of

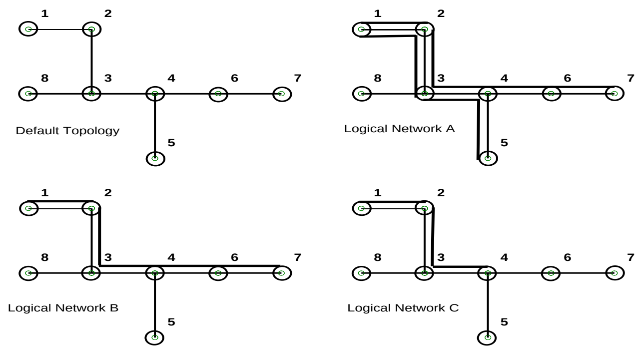

Figure 2.1: *Examples of default topology and logical networks*

hops in the shortest path on the physical network.

2. The cost to transport the traffic demands is proportional to the length of shortest path and the amount of traffic demand between a pair of nodes.

The cost for node i to be connected with each node $k \in S_i$ and carry traffic demand $d_{i,j}$ is defined:

$$C_i = \alpha \sum_{k \in B_i} h_{i,k} + \sum_{k \in S_i} t_{i,j} d_{i,j} \tag{2.1}$$

where B_i is the set of neighbors toward which node i has an overlay link with a neighbor node k, $h_{i,k}$ is the number of intermediate nodes in the physical path of $l_{i,k}$ and $t_{i,j}$ is the number of transit overlay links in the path to node j. The parameter α is a cost coefficient, which represents the relative weight of the two cost components: link creation cost and traffic transport cost. The total cost of the overlay network is consequently defined as:

$$C(G) = \sum_{i \in N} C_i \tag{2.2}$$

The cost model defined in the paper [7] and [16] is modified to include the traffic demand. It is important to note that C_i is a function of both the location of the overlay link $l_{i,j}$ and the demand $d_{i,j}$. Table 2.1 defines all the parameters composing the cost function. Figure 2.1 shows a simple example of an overlay network topology over a given physical network. For example, consider the default network in the figure where no overlay links are created, node 1 wants to send a traffic

Table 2.1: *Parameters and variables definitions*

Parameters	Definition
$h_{i,k}$	Number of intermediate nodes in the physical path $l_{i,k}$.
$t_{i,j}$	Number of transit overlay links in the path between node i and node j.
$l_{i,k}$	Number of hops in the shortest path between source node i and neighbor node k.
α	Overlay cost coefficient.
$d_{i,j}$	Traffic demand between node i and node j.
$a_{i,j}$	Element of the adjacent matrix equals to 1 if there is a physical link between node i and node j.
Variable	Definition
$\delta_{i,j}$	Binary decision variable equals to 1 if there is an overlay link between node i and node j.
$y_{i,j,k}$	Amount of flow leaving node i going to node j started from node k.

demand $d_{1,5}$ to node 5 and a traffic demand $d_{1,7}$ to node 7. If node 1 does not select any new neighbor node, it is only connected with node 2 and the cost for node 1 is only given by the routing cost. Since the number of links in the path from node 1 to node 5 equals to 4, the number of transit links to reach node 5 equals to 4-0-1=3 and to reach node 7 equals to 5-0-1=4, we have C_1= 3$d_{1,5}$+ 4$d_{1,7}$. In case of the overlay network A, node 1 selects nodes 5 and 7 as neighbors, so two overlay links are setup: one connecting node 1 with node 5 and the other connecting node 1 with node 7. The total cost is only given by the cost of creating the logical links. The second cost component related to the transport of the demands is zero, since no transit links are used because there are direct overlay links between the source node and the destination nodes. In this case we have C_1= 3α+ 4α.

Due to the different behaviors of the nodes in the network, we classify the problem formulation into two categories. One is the non cooperative (N-C node) behavior and the other is the cooperative (C node) behavior.

2.3.2 Integer Linear Programming

In this section, we present the ILP formulation of constructing the overlay topology given the traffic demands between the nodes. Two different node behaviors are considered.

1. C node: The new overlay link built between any two nodes can be used to route the traffic demands of other nodes.

2. N-C node: The new overlay link built by a given source can only be used by that source to route the traffic demand .

Consequently, the C node behavior implies the formulation of the global optimum while the N-C node implies the formulation of the local optimum for each source.

C node behavior

The decision variables used in this problem formulation are $\delta_{i,j}$ and $y_{i,j,k}$ where $\delta_{i,j}$ is the boolean decision variable of creating an overlay link between node i and node j and $y_{i,j,k}$ represents the amount of flow leaving node i going to node j started from source node k. Table 2.1 defines the decision variables and the parameters used in the formulation. The objective function is formulated as:

$$\begin{aligned} min \sum_{i\in N}\sum_{j\in N} 0.5\alpha h_{i,j}\delta_{i,j} \\ + \sum_{i\in N}\sum_{j\in N}\sum_{k\in N} y_{i,j,k} - \sum_{k\in N}\sum_{l\in N} d_{k,l} \end{aligned} \tag{2.3}$$

subject to:

$$\sum_{j\in N} y_{k,j,k} = \sum_{l\in N} d_{k,l} \; \forall \; k \tag{2.4}$$

$$\sum_{i\in N}(y_{i,j,k} - y_{j,i,k}) = d_{k,j} \; \forall \; k, j, k \neq j \tag{2.5}$$

$$\delta_{i,j} \geq a_{i,j} \; \forall \; i, j \tag{2.6}$$

$$\sum_{k \in N} y_{i,j,k} \leq M(\delta_{i,j} + a_{i,j}) \, \forall \, i,j \tag{2.7}$$

$$y_{i,j,k} \leq M(\delta_{i,j} + a_{i,j}) \, \forall \, i,j,k \tag{2.8}$$

Eqn.(2.3) shows the cost of creating an overlay link and the cost of routing the traffic demand. Eqns.(2.4-2.8) are the main constraints to the optimization problem; Eqn.(2.4) shows the total amount of the traffic demands sent by each source node; Eqn.(2.5) represents the balance of the coming and outgoing traffic demands through any node in the network; In Eqn.(2.6) we consider all the physical links are overlay links; Eqns.(2.7-2.8) show that the traffic demand can be routed on any new overlay link according to the shortest path between the source node and the destination node. These equations are called the link load equations [17] because the traffic demand on each link cannot exceed the link capacity. M is a large number which represents the incapacitated problem.

N-C node behavior

The C node problem formulation is a global optimization and the N-C node problem formulation can be reduced from the C node formulation as a local optimization. Each source node creates overlay links for its benefit to satisfy the demand volume to all its destinations. By repeating this process for each node in the network, the obtained overlay topology is the optimal overlay topology of the N-C node behavior. The final topology is the union of each source-multi destinations optimal topology. When reducing the C node formulation to the N-C node formulation we replace $\delta_{i,,j}$ with δ_j and replace both the source index i in $a_{i,j}$ and the source index k in $y_{i,j,k}$ and $d_{k,l}$ respectively with the source number. The problem formulation becomes,

$$\begin{aligned} min \sum_{j \in N} & 0.5 \alpha h_{source,j} \delta_j \\ & + \sum_{i \in N} \sum_{j \in N} y_{i,j,source} - \sum_{l \in N} d_{source,l} \end{aligned} \tag{2.9}$$

subject to:

$$\sum_{j \in N} y_{source,j,source} = \sum_{l \in N} d_{source,l} \tag{2.10}$$

$$\sum_{i \in N} (y_{i,j,source} - y_{j,i,source}) = d_{source,j} \; \forall j \neq source \tag{2.11}$$

$$\delta_j \geq a_{source,j} \; \forall \; j \tag{2.12}$$

$$y_{i,j,source} \leq M(\delta_j + a_{i,j}) \; \forall \; i, j \tag{2.13}$$

Algorithm 1 N-C node behavior

```
Adjacent Matrix=[]
for i = 1 to N do
    Run the C node formulation for source i
    Adjacent Matrix[i,:]=δ_j
end for
Generate the optimal overlay topology from the Adjacent Matrix
```

Algorithm 1 shows the generation of the optimal overlay topology for the N-C node behavior. The problem of creating overlay links in the network is *NP-hard* because it can be reduced to the Hamiltonian Path Completion problem which is in the *NP-complete* class [18].

2.4 Proposed Heuristics

In this section, we introduce different heuristics based on a greedy approach, Dijkstra's algorithm, node clusters, traffic demands and number of hops in the shortest path between node-pairs to generate near-optimal overlay topologies.

2.4.1 Heuristic 1: Greedy Heuristic A

In this heuristic each node compares the cost of creating a new direct overlay link with the cost of transporting the traffic demands using the existing overlay links in the network. Each node decides to create overlay links if this cost is less than the cost of transporting the traffic demands on the existing overlay network. The psuedocode is shown in algorithm 2.

Algorithm 2 Greedy Algorithm A

for $i = 1$ to N **do**
  for $j = 1$ to N **do**
    if $j \neq i$ **then**
      $OverlayCost$=$\alpha\ h_{i,j}$
      $TransportCost$=$t_{i,j}\ d_{i,j}$
      if $OverlayCost \leq TransportCost$ **then**
        Create an overlay link between nodes i and j
      end if
    end if
  end for
end for
Get the shortest paths
Compute the overall cost

2.4.2 Heuristic 2: The Dijkstra Heuristic

The second algorithm starts with a fully mesh overlay network. The link weights of the fully mesh network are computed according to the characteristics of the cost function in (2.1). In particular, the link weight is computed as follow

$$Weight_{i,j} = \frac{\alpha}{t_{i,j} d_{i,j}} \tag{2.14}$$

It represents some common situations in designing the overlay topology. For decreasing α, the link weight decreases because the cost of creating an overlay link decreases. For increasing traffic demand $d_{i,j}$, the link weight increases which means that it is not worth to create an overlay link to transport small traffic demands.

The heuristic is based on the following steps: a node i is randomly chosen and Dijkstra Algorithm is applied to obtain the Dijkstra shortest path tree from the chosen node toward all the destinations. The tree is added to the previous feasible solution (initial feasible solution is the underlay network) and the total cost is computed. The new solution is accepted only if the current cost is less than the previous cost. The heuristic terminates if it can not find a lower cost solution after M iterations where M is a large number.

Algorithm 3 The Dijkstra Heuristic

Construct a fully mesh overlay network
Compute link weights
$Weight_{i,j}$=α /($t_{i,j}$ $d_{i,j}$)
$Count$=0
K=0
$T_{BestAdjacent}$=$T_{UnderlayAdjacent}$
while $Count < M$ **do**
  K=K+1
  Randomly choose a node i from the network
  Apply Dijkstra Algorithm given the source node i
  $T_{Dijkstra}$=Topology of the Dijkstra tree
  T_{Temp}=$T_{BestAdjacencyK-1} \cup T_{Dijkstra}$
  t=shortest paths of T_{Temp}
  C_{Temp}=Cost of the topology T_{Temp}
  if $C_{Temp} > C_{K-1}$ **then**
    $Count$ = $Count$+1
    $Cost_K$=$Cost_{K-1}$
    $T_{BestAdjacencyK}$=$T_{BestAdjacencyK-1}$
  else
    C_k=C_{Temp}
    if C_{Temp}=C_{K-1} **then**
      $Count$=$Count$+1
    else
      $Count$=0
    end if
    $T_{BestAdjacencyK}$=T_{Temp}
    Update $LinkWeight_{i,j}$
  end if
end while

For each value of α, number of iterations is computed to judge the speed of the convergence of the heuristic.

2.4.3 Heuristic 3: Greedy heuristic B

This heuristic is different from heuristic 1 in that a sequence of nodes is selected. The first node selects the best neighbor to minimize its incremental cost and establishes a new overlay link. The next node in the sequence also selects the best neighbor node, taking into account the previously established overlay links if nodes are C-node.

2.4.4 Heuristic 4: Node Clustering heuristic

The shortest path between any source-destination pair contains nodes with high node degree on it. In this heuristic, nodes in the network are grouped in a decentralized way. In each group, there is a leader node which has high node degree. We define a relay node, which is the nodes physically connected with more than one leader node in the network. Ordinary nodes are the remaining nodes in the group. The leader nodes in the network establish direct overlay links between them. In order to create the groups and select the leaders, we propose the following decentralized procedure. Each node i sends information about its node degree to the physical neighbors and it receives their node degree information. If a given node has the highest node degree among its neighbors, it will consider itself a leader node. If not, it may be either a relay node or an ordinary node. If node i is a leader node, it informs all its physical neighbors that it becomes the leader of the group. If any ordinary node receives at least two messages from different leader nodes, it will consider itself as a relay node, it selects randomly one leader and it will begin to inform its neighbors about the selected one. If an ordinary node does not receive information from any leader node, it selects the neighbor node with the maximum node degree and joins its group. Each leader node in the network maintains a list of all the leader nodes in the network. When a leader node receives information about a new leader in the network, it saves it in its leader nodes list. Using this list, each leader node runs the C node optimization program to decide about the new overlay neighbor nodes toward which it builds overlay links.

Algorithm 4 The Node Clustering Algorithm

For each node i in the network.
ND_i: Node Degree of node i.
$NDN_{j,i}$: A matrix saved at node j containing the node degree of the neighbor node i.
LN: Leader Node.
RN: Relay Node.
NLN_i: Number of Leaders that node i is physically connected.
Collecting the node degrees of the neighbors.
for $j = 1$ to N **do**
  if $a_{i,j} == 1$ **then**
    $NDN_{j,i} = ND_i$
  end if
end for
for $j = 1$ to N **do**
  $NDN_{i,j}$=ND_j
end for
Choosing the leader nodes
Max Degree=$max(NDN_i)$
if $max(NDN_i)$ = = ND_i **then**
  Node i=LN
  for $j = 1$ to N **do**
    if $a_{i,j}$ = = 1 **then**
      My Leader=i
    end if
  end for
else
  if $NLN_i >= 1$ **then**
    Node i=RN
  else if $NLN_i == \phi$ **then**
    My Leader = My Leader($max(NDN_i)$)
  end if
end if

Table 2.2: *Characteristics of the underlay topologies*

	USIP	RF1	RF2
Number of nodes n	24	35	112
Number of links m	43	79	147
Average node degree $\bar{K}$	3.5833	4.5143	2.6250
Max. node degree $max(K)$	5	26	25
Power law exponent γ	-	1.6	2.45
Diameter D	6	4	6
Assortativity r	0.1497	-0.4527	-0.3982

2.4.5 Heuristics 5,6 and 7: Max-Length, Max-Demand and Max-Length-Demand

From the cost function characteristic eqn.(2.1), it is evident that establishing overlay links toward far destinations and/or carrying high traffic volumes is economically advantageous. Based on these motivations, we propose the following heuristics where each node establishes an overlay link with respectively maximum distance destination $max(l_{i,j})$, maximum traffic demand destination $max(d_{i,j})$ and maximum distance-traffic demand combination destination $max(l_{i,j}d_{i,j})$. If the source node finds more than one destination with the same maximum decision parameter, it randomly chooses one and builds with it an overlay link. Finally, each node informs its physical neighbors to update the shortest paths to all their destinations if nodes are C-node.

2.5 Underlay Networks, Topology Characteristics and Traffic Demand Matrices

2.5.1 Underlay Networks

The ILP formulations and the heuristics are applied to a 24-node network representing a US nation-wide IP backbone network topology [19], a 35-node and a 112-node Rocketfuel network topologies [20]. The characteristics of each underlay network are shown in table 2.2.

2.5.2 Topology Characteristics

Some topology characteristics shown in [21], [22] and [23] are used to analyze the generated optimal and near-optimal overlay topologies.

Average Node Degree $\bar{k}$

The average node degree is defined as $\bar{k} = 2m/n$ where m is the total number of links and n is the total number of nodes in the topology. The average node degree measures the overall connectivity of the generated topology.

Node Degree Distribution NDD

It is the distribution of the node degrees in the network. $P(K)$ is the probability that a node has a node degree of K where $P(K) = n(k)/n$. The power law of the degree distribution is defined as $P(K) \sim K^{-\gamma}$ where γ is the power law exponent.

Assortativity Coefficient r

Assortativity coefficient ($-1 \leq r \leq 1$) reflects the proportion between the radial links which connect nodes with different node degrees and the tangential links which connect nodes with similar node degrees. Networks are either assortative ($r \geq 0$) where number of tangential links are greater than number of radial links. Assortative networks are immunized from fast spread of viruses. The opposite properties are applied to disassortative networks. To compute the assortativity coefficient, the joint degree distribution JDD has to be computed first. The joint degree distribution is defined as $P(K_1, K_2) \sim m(K_1, K_2)/m$.The joint degree distribution $P(K_1, K_2)$ represents the probability that a selected link from the topology connects two nodes with node degrees K_1 and K_2. The exact mathematical form to compute the assortativity coefficient r could be found in [24].

Diameter D

The Diameter D is the maximum shortest path in the topology. It reflects the overlay reachability of the farthest nodes in the underlay network.

Clustering Coefficient c

The clustering coefficient c of node i is the ratio between the existing number of links interconnecting the neighbors of node i in the topology and the required number of links to fully interconnect the neighbors. It represents the local robustness for each node in the network.

2.5.3 Traffic Demand Scenarios

Diffrent traffic demands are used in simulations. The traffic scenarios are

- Homogeneous traffic demand: The traffic demands of all the node-pairs are the same.
- Uniform traffic demand: Random traffic demand following the uniform distribution between 0 and 24.
- Bimodal traffic demand: Random traffic demand following the bimodal distribution with coefficient of variations CVs of (0.125, 0.05) as given in [11] and mean values of 8 and 20.

Each of the uniform and the bimodal traffic demand scenarios allows a high level of variety in the traffic demands between the nodes in the network.

2.6 Results and Discussion

The ILP formulation, which provides optimal overlay topologies, and the heuristics are applied to the network topologies discussed in section 2.5 given the C-Node and N-C Node behavior of nodes. Extensive testing and simulations are done on the heuristics to compare the generated topologies with the optimal ones. The generated topologies are deeply analysed to understand the effect of the traffic demands, overlay cost coefficient and the underlay topology on the created overlay topology.

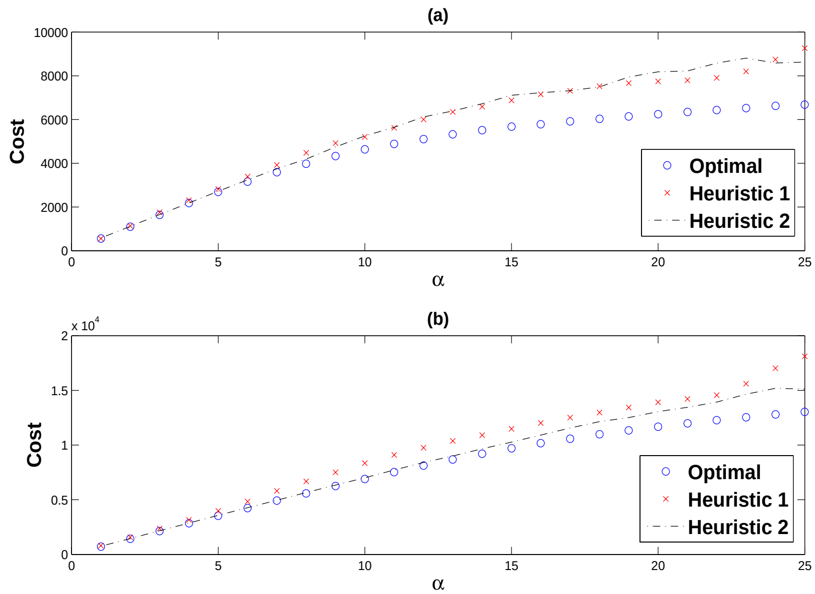

Figure 2.2: *The optimal overlay topology cost for* $1 \leq \alpha \leq 25$ *for a) 24-node network b) 35-node network*

2.6.1 Results: Part 1

This section addressed the results collected from the ILP formulation and Heuristic 1 and Heuristic 2 in case of the C-Node behavior of nodes with bimodal traffic scenario and the three different underlay topologies discussed in section 2.5.

Integer Linear Programming

Overlay Network Cost

The overall cost function of the optimal overlay topologies are computed given different values of α. Figure 2.2 shows the topology cost in the range of α where the optimal solution were obtained. The optimal topologies obtained for the 24-node network have been analysed. The same analysis is also applied for the 35-node network. The observations are as follow:

Average Node Degree, Node Degree Distribution and Joint Degree Distribution

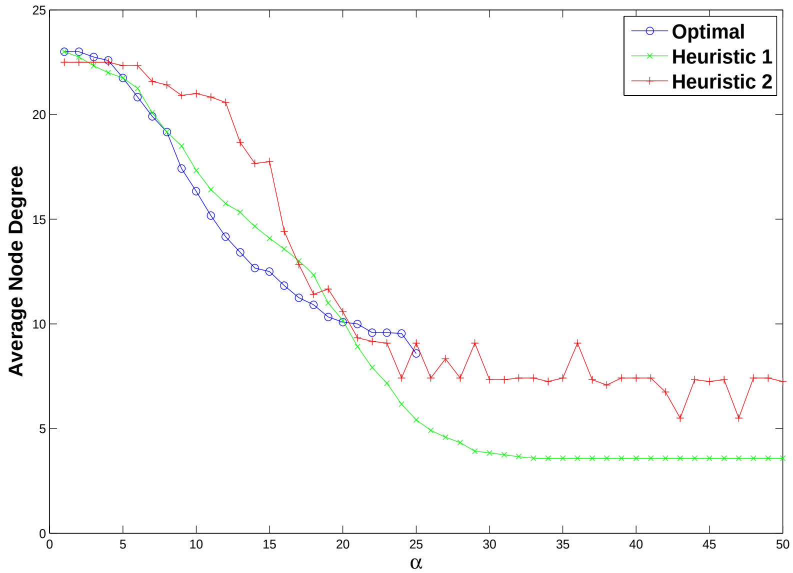

Figure 2.3: *Average Node Degree for the optimal and near-optimal topologies with different values of α for the 24-node underlay network*

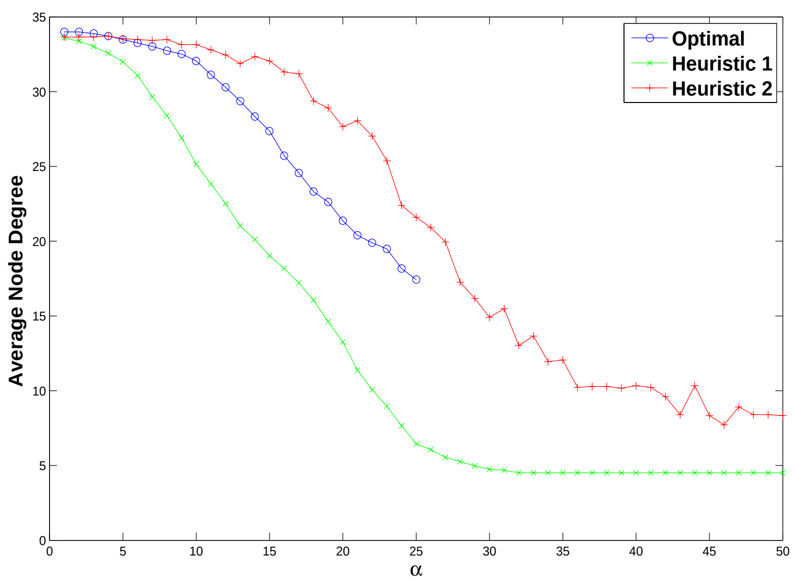

Figure 2.4: *Average Node Degree for the optimal and near-optimal topologies with different values of α for the 35-node underlay network*

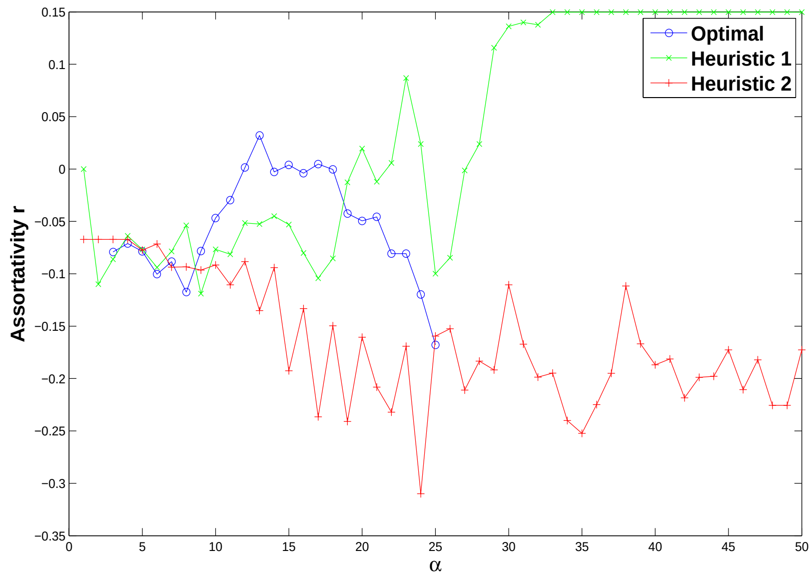

Figure 2.5: *Assortativity coefficient r for the optimal and near-optimal topologies with different values of α for the 24-node underlay network*

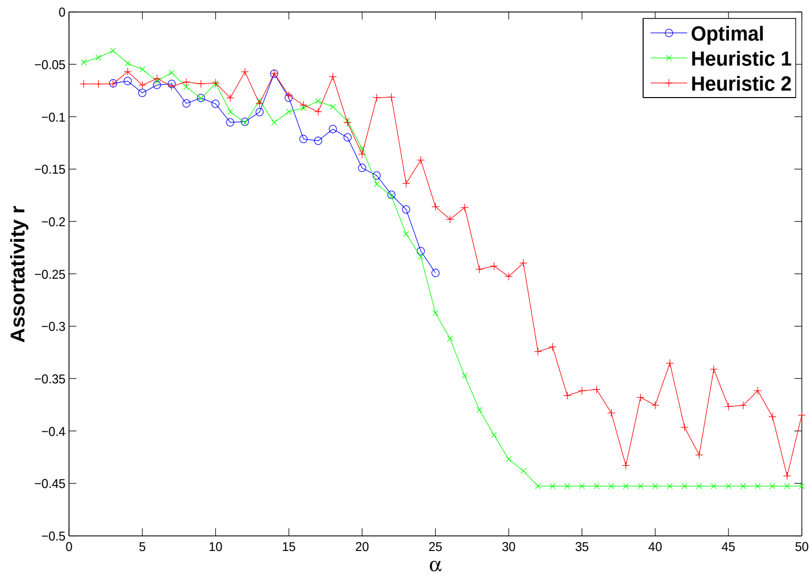

Figure 2.6: *Assortativity coefficient r for the optimal and near-optimal topologies with different values of α for the 35-node underlay network*

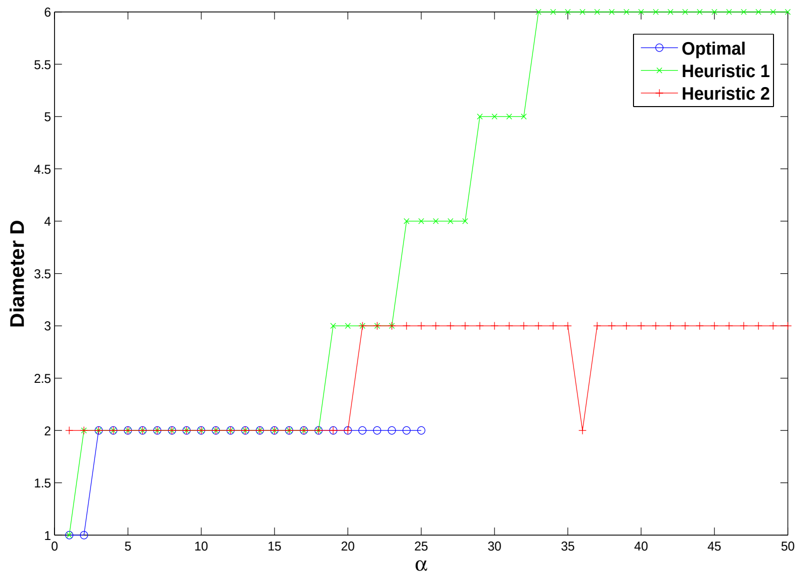

Figure 2.7: *Diameter D for the optimal and near-optimal topologies with different values of α given 24-node network*

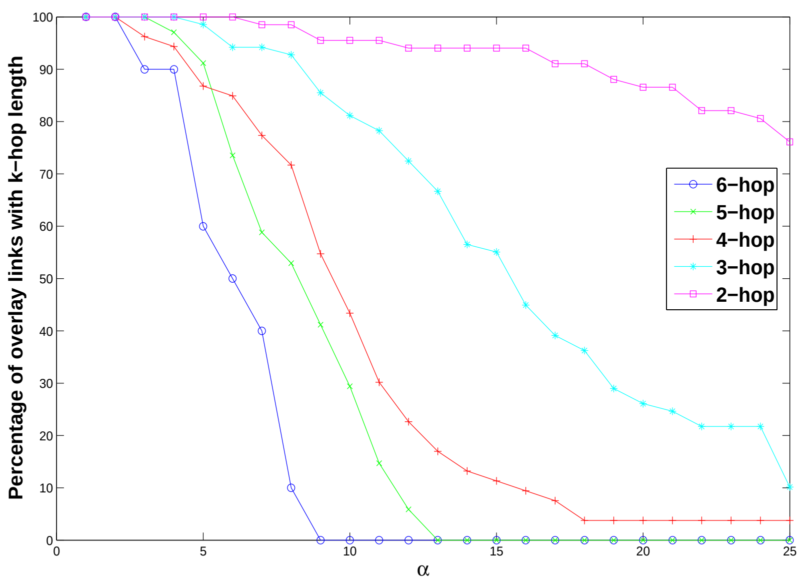

Figure 2.8: *Percentage of overlay links with k-hop length in the optimal topologies for the 24-node underlay network*

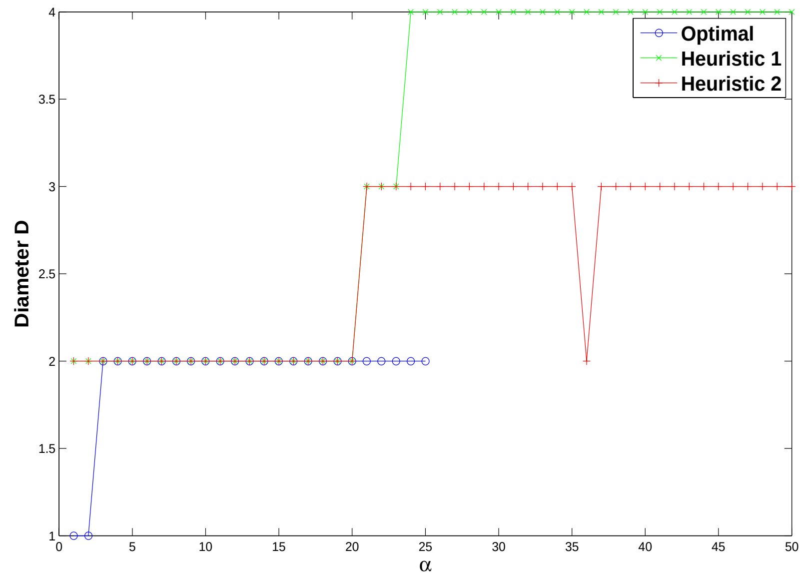

Figure 2.9: *Diameter D for the optimal and near-optimal topologies with different values of α for the 35-node underlay network*

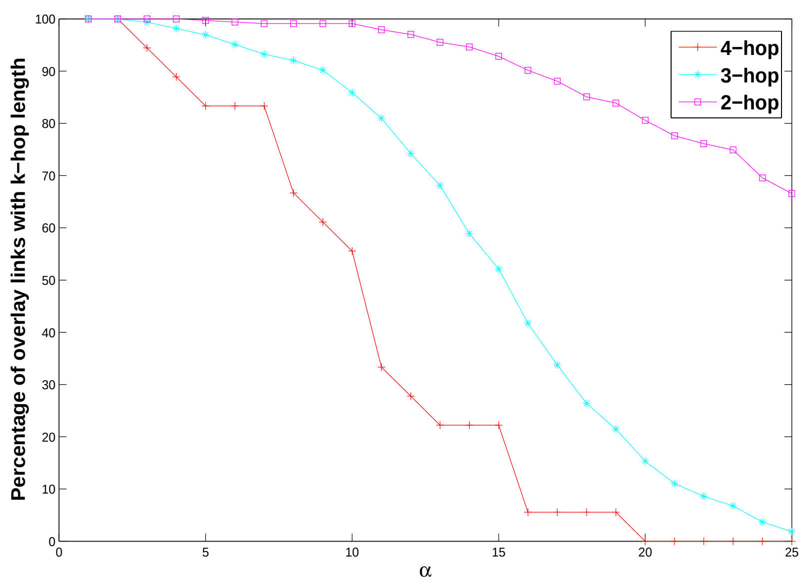

Figure 2.10: *Percentage of overlay links with k-hop length in the optimal topologies for the 35-node underlay network*

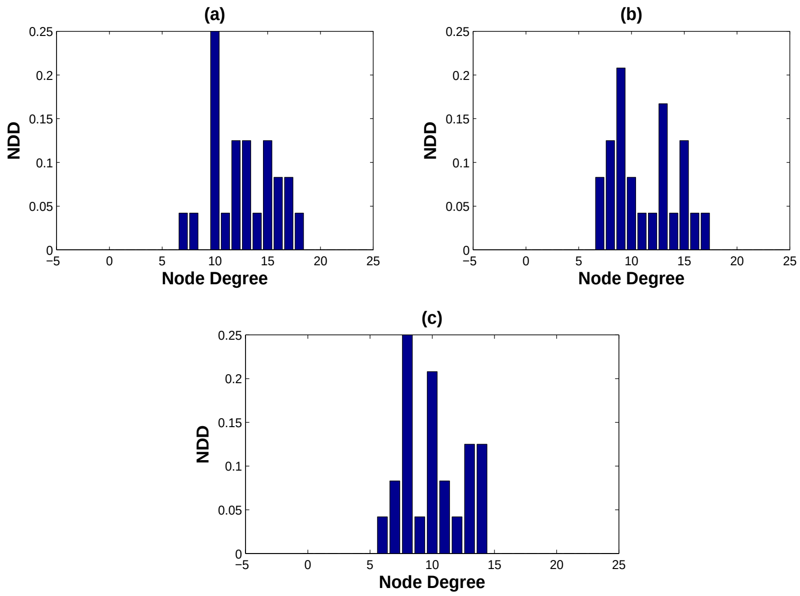

Figure 2.11: *a)* $\alpha = 14$*, maximum average clustering coeffient is 0.85 at node degree 8. b)* $\alpha = 17$*, maximum average clustering coeffient is 0.755 at node degree 8. c)* $\alpha = 20$*, maximum average clustering coeffient is 0.725 at node degree 6.*

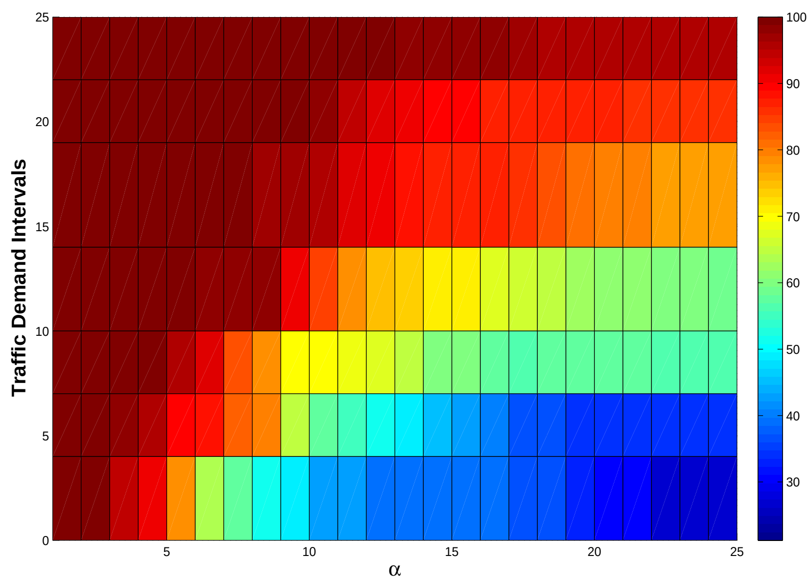

Figure 2.12: *Percentage of overlay links connecting two nodes with traffic in the demand interval for the 24-node underlay network*

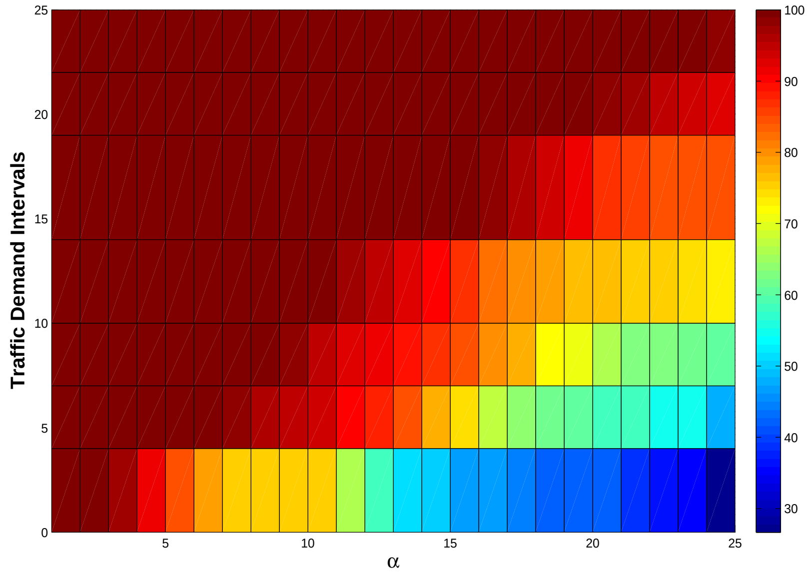

Figure 2.13: *Percentage of the overlay links connecting two nodes with traffic in each demand interval for the 35-node underlay network*

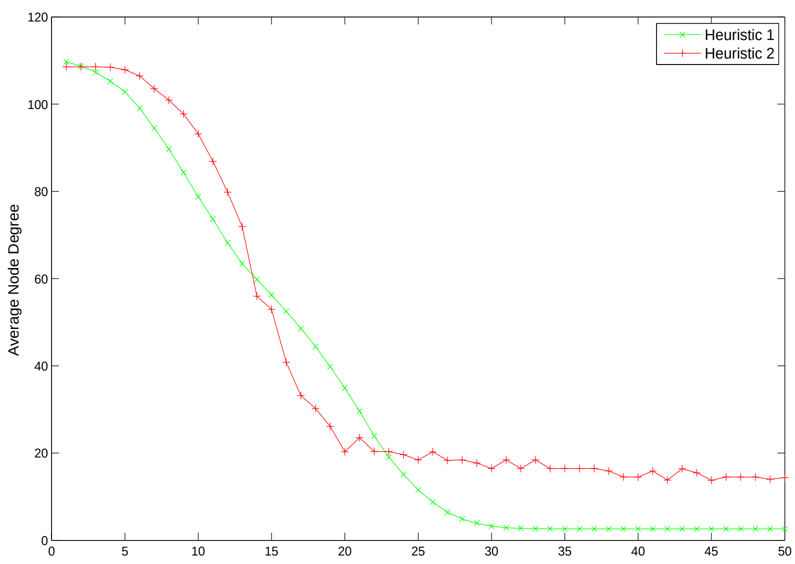

Figure 2.14: *Average Node Degree of the near-optimal topologies with different values of α for the 112-node underlay network*

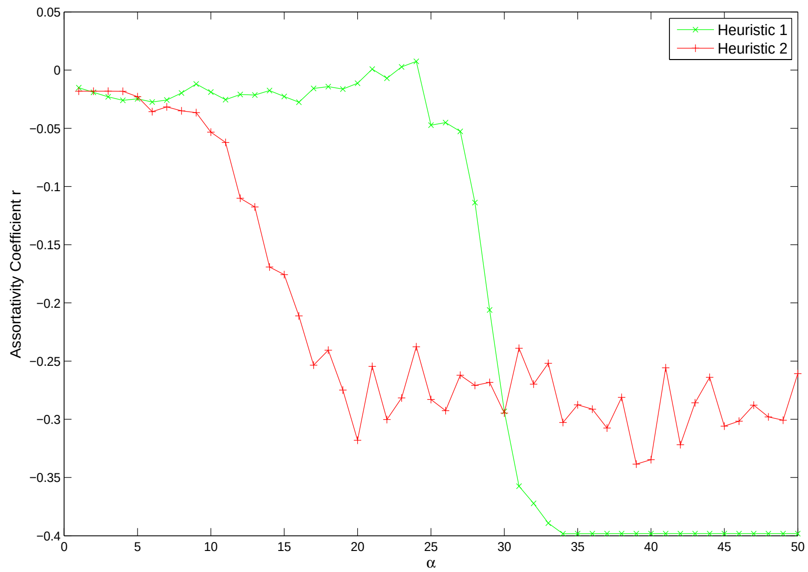

Figure 2.15: *Assortativity coefficient r of the near-optimal topologies with different values of α for the 112-node underlay network*

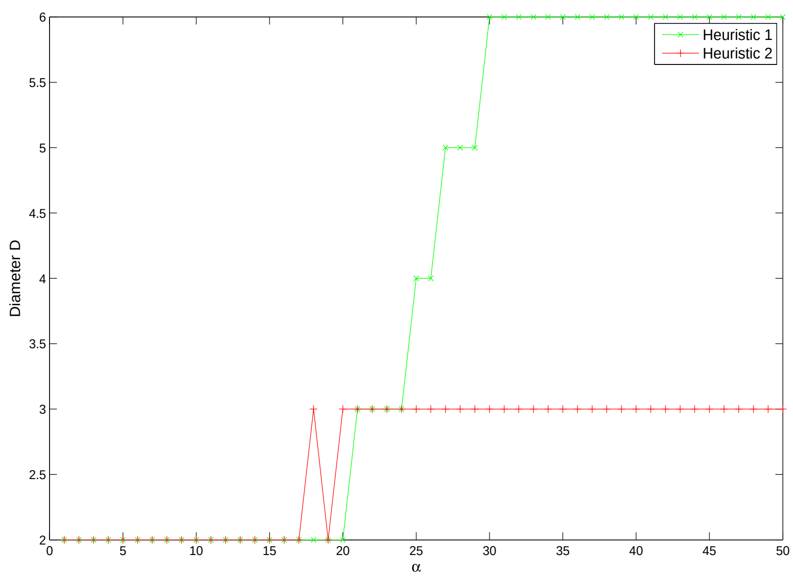

Figure 2.16: *Diameter D of the near-optimal topologies with different values of α for the 112-node underlay network*

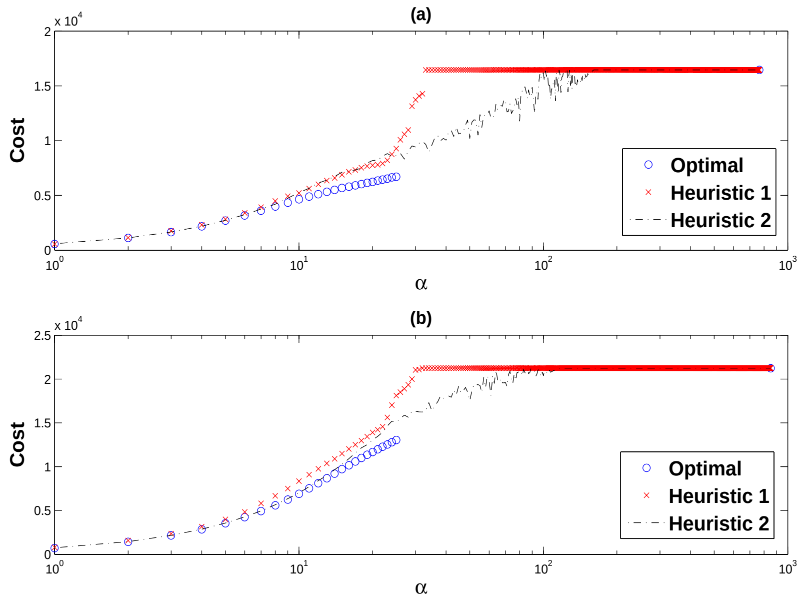

Figure 2.17: *The optimal overlay topology cost with* α *for a) 24-node network b) 35-node network*

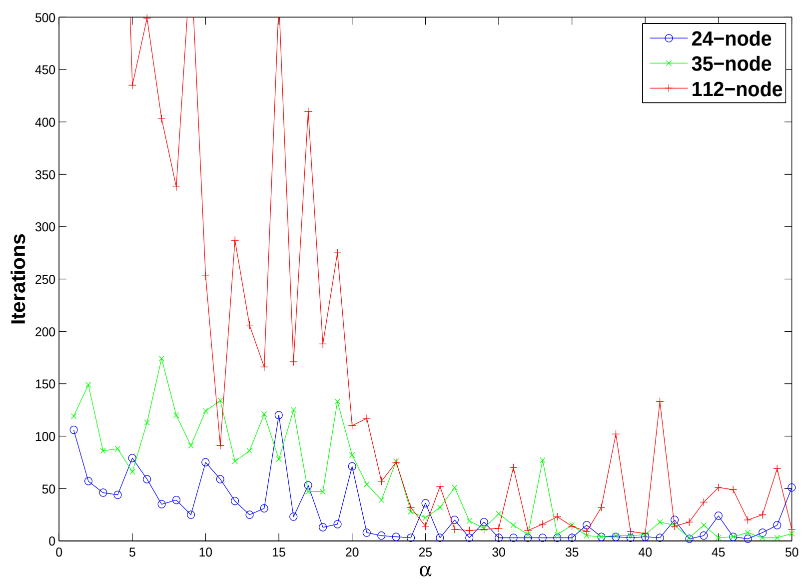

Figure 2.18: *Number of iterations of heuristic 2 for the three underlay topologies*

Figure 2.3 shows the average node degree of the optimal topologies. Given small value of α (α=1 and 2), the fully mesh network is the optimal topology. Increasing α, the optimal topology becomes less dense. In particular, for α equals 3 to 6, lower node degrees (referred to the full node degree =23) appear in the topologies and the probability that those nodes with lower node degrees are connected with nodes having higher node degrees increases. They do not have the incentive to be connected together, making the obtained topologies disassortative.

For α equal to 6 and 7, some node-pairs with traffic demands greater than twice α drop overlay links connecting them. To minimize the overall cost, these nodes are connected with nodes having high node degrees to shorten the path toward the destinations.
For α equal to 8 and 9, nodes with full node degree (23) disappear from the topologies. The topologies are more disassortative.
When α is equal to 10, nodes with the lowest node degrees have the incentive to be not only connected with the highest degree nodes but also connected together. As α increases, the probability that nodes with the highest node degrees are interconnected decreases and the probability that nodes with low degree are interconnected increases.
Assortativity Coefficient

One of the most important network characteristic is the assortativity coefficient r. The assortativity of the 24-node underlay topology is 0.1497 which means that it is assortative because of the uniformly distribution of the node degree.On the other hand, due to the exponentially distributed node degree of the 35-node topology, the network is disassortative with assortativity coefficient -0.4527. Figures 2.5 and 2.6 show the assortativity coefficient of the optimal overlay topologies. From these figures we can see the effect of the underlay topology on the optimal overlay topology. Networks with node degree exponentially distributed have many nodes with small node degree while they have few nodes with high node degree. Many node-pairs have common parts of the shortest path, which results in the dependence of many nodes on the cooperative behavior to get free rides instead of creating direct overlay links. From the JDD of the optimal topologies, we found that nodes with small node degrees are not interconnected via overlay links but connected with nodes with high node degrees keeping the generated topologies disassortative.

Diameter

Figure 2.7 shows the diameter as a function of α for the optimal topologies and for topologies obtained using heuristics 1 and 2. For the optimal topologies, the diameter is equal to 1 for α=1 and 2, since the network is fully connected. When α increases, the network becomes less dense and the diameter is 2.

An explanation for this behavior can be also obtained from figure 2.8. In this figure, the length of the overlay links in the optimal topologies is considered. The length is measured in terms of number of hops (k-hop) in the underlay network. The maximum length of an overlay link is equal to 6 which is the diameter of the underlay 24-node network. Considering the fully connected network obtained for α=1 and 2, the overlay links can be classified based on their lengths in the range 2 to 6.

In figure 2.8, the percentage of overlay links with k-hop length in the optimal topologies is shown as a function of α. From α=9, all the overlay links with length 6-hop are not created in the optimal topologies. From α=13, only overlay links with 2, 3 and 4-hop lengths are parts of the optimal topologies. Similar analysis can be performed for the 35-node network as shown in figures 2.7 and 2.10 for the diameter and percentage of overlay links respectively.

Clustering Coefficient

For each degree of nodes in the optimal overlay topology, the average clustering coefficient is computed between nodes with that node degree. The relationship between the average clustering coefficient, node degree distribution and alpha is observed. For the intervals of α, $3 < \alpha < 9$, $12 < \alpha < 15$ and $20 < \alpha < 25$, the node degree with the largest average clustering coefficient has the minimum node degree probability while for the intervals of $alpha$, $10 < \alpha < 11$ and

$16 < \alpha < 19$, the above phenomenon is not observed.

We explain the inconsistent behavior of the node degree distribution with the average clustering coefficient as follow: For $\alpha = 1$ and 2, the optimal overlay topology is the fully mesh network where each node has the maximum node degree with probability of one and the corresponding average clustering coefficient is also one. As α increases, the maximum node degree of the optimal overlay topology (refered to the full node degree) decreases causing the redistribution of the node degree with the appearance of lower node degrees in the network. Nodes with the largest average clustering coefficient have the minimum probability of node degree as shown in figure 2.11a. When α increases further, some nodes with node degrees smaller than the node degree of those with largest average clustering coefficient, appear in the topology. Their number is smaller than those with largest average clustering coefficient as shown in figure 2.11b. As α increases, number of nodes with smaller node degrees appears in the overlay topology with low probability and their neighbors are strongly interconnected and have the largest clustering coefficient as shown in figure 2.11c.

Traffic Demand and α

The traffic spectrum is grouped in intervals to reduce the complexity of the analysis. The intervals are: $1 \leq d1 \leq 3$, $4 \leq d2 \leq 6$, $7 \leq d3 \leq 9$, $10 \leq d4 \leq 13$, $14 \leq d5 \leq 18$, $19 \leq d6 \leq 21$, $22 \leq d7 \leq 25$ and $26 \leq d8$. The percentage of overlay links connecting node-pairs with traffic demands within the given traffic intervals is observed. We found that nodes with traffic demands greater than twice α, have a high chance to be directly connected via overlay links.

Figure 2.12 describes the generated optimal topologies in terms of the traffic demands. The figure represents the percentage of overlay links connecting node-pairs with traffic demands belonging to the given traffic intervals. For α equal to one and two, all possible connection are in the topologies since the topologies are fully mesh networks. It means that the cost of creating overlay links is very low comparing to the transport cost. As α increases, the traffic demands at which node-pairs connected via a given percentage of overlay links, increases. For a given traffic demand interval

(horizontal view) and for small values of α, all the corresponding node-pairs are connected directly via overlay links. As α increases, only a decreasing percentage of the corresponding node-pairs are connected by direct overlay links.
For a given α (vertical view), for the interval with maximum traffic demands, almost all the corresponding node-pairs are connected via direct overlay links. For decreasing traffic demands, only a percentage of the corresponding node-pairs are connected by direct overlay links. From a qualitative evaluation, the isopercentage curves are concave functions.

The above observations are extended for the 35-node overlay network for the same topology metrics.

Running time

The running time T (in dd:hh:mm:ss) to solve the ILP problem is summarized as follow:
For 24-node network T=00:00:03:40 for α=10, and T=03:02:08:54 for α=24. For 35-node network T=00:00:05:30 for α=10, and T=02:01:01:55 for α=24.
From the above analysis, the optimal overlay topology becomes less dense as α increases until the optimal overlay topology becomes the underlay topology. The minimum value of α at which the optimal topology is the underlay topology, is called $\alpha_{threshold}$. For $\alpha >= \alpha_{threshold}$, the optimal overlay topology remains the underlay network.

We believe that there exists an interval of α ($\alpha_x <= \alpha < \alpha_{threshold} - 1$) where the optimal overlay topologies will have one link in addition to the default topology. Algorithm 5 represents the procedure of the exhaustive search to compute $\alpha_{threshold}$. Starting with an initial value of α and increasing α by one in each step, a pair of nodes which are not connected is selected and it is assumed that there is an overlay link connecting them. The overall cost is computed and the algorithm is iterated until the overall cost is greater than or equal to the cost of the default topology. We also believe that the problem of finding $\alpha_{threshold}$ could be constrained by only choosing a pair of nodes which are separated by two hops in the default topology.

Algorithm 5 Exhaustive Search to compute $\alpha_{threshold}$

MinCost=0
α = InitialValue
while $MinCost < DefaultCost$ **do**
  TempAdj = Adjacency
  k=0
  α = α + 1
  for $i = 1$ to N **do**
    for $j = i + 1$ to N **do**
      if $Adjacency_{i,j} == 0$ **then**
        k=k+1
        $TempAdj_{i,j} = 1$
        $TempAdj_{j,i} = 1$
        $TempCost_k$ = Compute the overall cost
      end if
    end for
  end for
  $MinCost$ = min ($TempCost$)
end while
$\alpha_{threshold}$=α

The obtained results with constraining number of hops between the selected pair of nodes are the same as the obtained results without constraining number of hops between the selected pair of nodes for the 24-node, 35-node and 112-node networks.

The node degree of the selected pair of nodes at which the optimal overlay topology is found at $\alpha = \alpha_{threshold} - 1$ are observed. We found that in the 24-node network, the selected nodes have node degrees of 3 and 5 which are around average and maximum node degrees respectively, while in the 35-node network, the selected nodes have node degree of 1 and 26 which means that nodes with minimum and maximum node degrees are selected.

The traffic demands between the pair of nodes are observed. For the 24-node network, the selected nodes have the maximum possible traffic demands in the network while for the 35-node network, the selected nodes have traffic demands around the averages of the bimodal traffic in the network. The algorithm is applied to the 112-node network and we found that the traffic demands between the selected nodes are around the averages of the bimodal traffic in the network.

This analysis will help to identify $\alpha_{threshold} - 1$ and consequently $\alpha_{threshold}$ by a proper selection of the pair of nodes which has the only additional overlay link based on their traffic demands and their topological characteristics in the network.

Heuristics

Heuristics are applied to the three different network sizes shown in 2.5.

Overlay Network Cost

Figure 2.17 shows the cost of the optimal overlay topologies compared with the cost of the obtained near-optimal topologies generated by the heuristics. The figure shows that the heuristic solutions have costs close to the costs of the optimal solutions for $1 \leq \alpha \leq 25$. For $\alpha > 25$, heuristic 2 solutions have costs lower than the corresponding costs of heuristic 1 solutions. Therefore, heuristic 1 can be used to find the near-optimal overlay topologies in the range of α up to 25 because it is simpler than heuristic 2, while heuristic 2 can be used for the range of $\alpha > 25$.

Average Node Degree

The average node degrees of the generated near-optimal overlay topologies follow the average node degree of the optimal topologies for the different values of α as shown in figure 2.3. For the 35-node network, heuristic 2 follows the optimal average node degrees for $1 \leq \alpha \leq 6$. For $\alpha > 6$, heuristic 2 generates denser overlay topologies than those generated by heuristic 1.

Assortativity Coefficient

Topologies generated by heuristic 1 follow the optimal topologies in terms of the assortativity coefficient for the 24-node network as shown in figure 2.5. For high values of α, r is positive reflecting the effect of the underlay topology on the overlay topology. The assortativity coefficient of the generated topologies by heuristic 2 for large value of α approaches the assortativity values of the optimal topologies for the same values of α. Heuristic 2 generates disassortative topologies regardless the values of α and the network size. In case of 35-node network, heuristic 1 generates overlay topologies with assortativity coefficient values approaching the optimal values as shown in figure 2.6.

Diameter

Figures 2.7 and 2.9 represent the change of the overlay topology diameter with α. For $\alpha = 1$ and 2, both heuristic 1 and heuristic 2 generate overlay topologies with a different diameter compared to the optimal topology diameter. Heuristic 1 and 2 generate overlay topologies with the same diameters as the diameters of the optimal topologies for α between 3 and 20 with $D = 2$ for both the 24-node and the 35-node networks. For $\alpha > 20$ and for both 24-node and 35-node networks , heuristics 1 and 2 generate topologies with higher D to reduce the addition of expensive overlay links.

Figures 2.14, 2.15 and 2.16 show the results of the heuristics when applied to the 112-node network. The average node degrees of the overlay topologies generated by heuristic 1 and heuristic 2 decrease smoothly as α increases. From the assortativity curve, heuristic 2 generates dissasortative topologies, while heuristic 1 generates both assortative and disassortative topologies based on the value of α.

Heuristic 1 generates topologies with all possible diameters D while heuristic 2 generates topologies with diameter of 2 and 3.

The convergence of heuristic 2 is shown in figure 2.18 for the three different network topologies. As the network size increases, the number of iterations increases too. The number of iterations decreases as α increases because the cost of creating overlay links becomes expensive when α increases. Heuristic 2 does not have the incentive to create many overlay links with high values of α. The generated topologies have low density so heuristic 2 convergence quickly to a less dense topology.

From the convergence behavior of heuristic 2 and the cost comparison among heuristics 1 and 2 and the optimal solution, heuristic 1 is recommended for $1 \leq \alpha \leq 25$, while heuristic 2 is recommended for $\alpha > 25$.

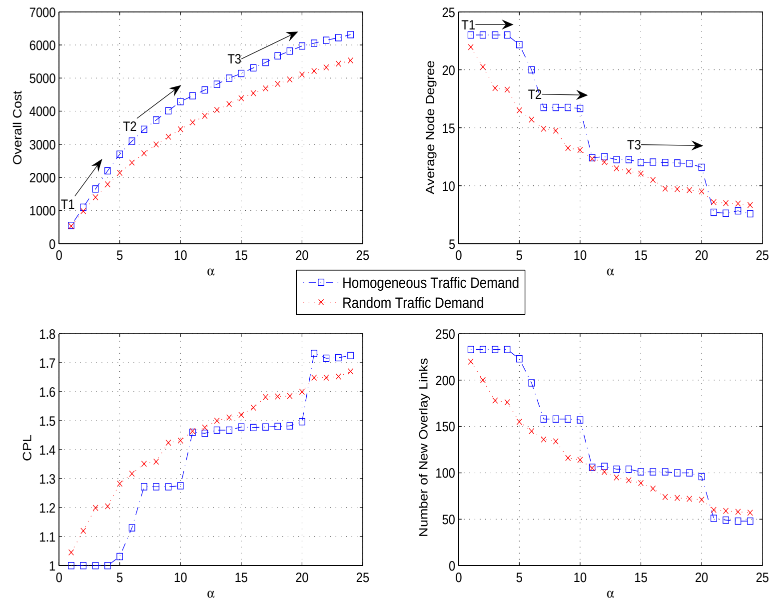

Figure 2.19: *Overall network cost, average node degree, characteristic path length and number of new overlay links for different values of α in case the N-C node behavior for both the random and the homogeneous traffic matrices*

2.6.2 Results: Part 2

The ILP formulation of the C-Node and N-C Node behavior of nodes which provide optimal overlay topologies and the heuristics are applied to the 24-node network discussed in section 2.5. Two traffic scenarios matrices are used 1) homogeneous traffic matrix 2) random traffic matrix. We compute the network costs and some graph metrics characterizing the generated topologies.

Integer Linear Programming

N-C node behavior

Figure 2.19 shows the overall network cost and some metrics graph characterizing the generated optimal overlay topologies. When the traffic demand matrix is homogeneous, few optimal overlay topologies are found for α intervals. For this reason, the graph metrics in those intervals are constant. For example, when $1 < \alpha \leq 4$, the optimal topology (T1) is the fully connected network. When $7 < \alpha \leq 10$, the optimal topology (T2) is a less connected graph and the average

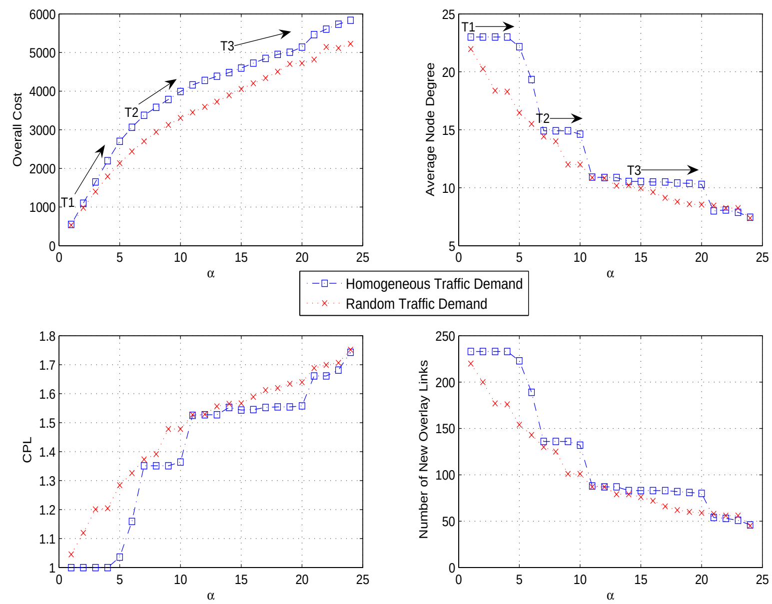

Figure 2.20: *Overall network cost, average node degree, characteristic path length and number of new overlay links for different values of* α *in case the co operative behavior of the nodes for both the random and the homogeneous traffic matrices*

node degree is constant and equal to 16.5. When the traffic demand matrix is random, the overall cost increases smoothly. When α is very small ($1 < \alpha \leq 2$), the optimal overlay topology is very close to the fully connected network. As α increases, the topology becomes less dense approaching the default topology.

C node behavior

Figure 2.20 shows the overall network cost and some graph metrics characterizing the generated optimal overlay topologies. When the traffic demand matrix is homogeneous, few optimal overlay topologies are found for some intervals of α, similar to the intervals found in N-C node behavior results. The results show that the network cost of the N-C node is higher than the network cost of the C node. The average node degree of the N-C node and number of new overlay links are higher than those of the C node. When α is very small, the optimal overlay topologies of the N-C node and the C node behaviors are similar for both the homogeneous and the random traffic matrices. As α increases, the optimal overlay topology of the N-C node is more dense than the optimal overlay topology of the C node. In the N-C node behavior, the source nodes build many overlay links to minimize the overall cost, while in the C node behavior, the source nodes don't build many overlay links, since they can use new overlay links built by other nodes.

Running time

Again, the running time T (in hh:mm:ss) to solve the ILP problem is summarized as follow:

- N-C node behavior: For homogeneous traffic demand T=00:01:30 for α=10 and T=00:07:50 for α=24. For random traffic demand T=00:01:28 for α=10 and T=00:10:17 for α=24.

- C node behavior: For homogeneous traffic demand T=00:10:40 for α=10 and T=03:08:54 for α=24. For random traffic demand T=00:03:40 for α=10 and T=01:01:55 for α=24.

Obviously, the running time of the C node problem is much greater than the running time of the N-C node problem. Therefore, in the following section, we apply our heuristics to solve the optimization problem for the C-node behavior. Clearly, when the size of the problem increases (number of nodes n), our heuristics will be needed to solve the N-C node optimization problem too.

Heuristics

Heuristics 3, 4, 5, 6 and 7 are compared with the ILP results. For the C node behavior, The ILP C node cost curve represents the lower bound for any topology and for any value of α as shown in Figure 2.21. When the traffic demand matrix is homogeneous, heuristic 3 and the ILP results are the same for small values of α. As α increases, the greedy heuristic is still the best heuristic but not the same as the ILP results. When α is greater than twice the value of the homogeneous traffic demand, heuristic 5 is the best. When the traffic matrix is random, heuristic 3 is the best and approaches the optimality up to α equal to the maximum traffic demand. As α increases heuristic 6 becomes the best one. The default topology is the solution for heuristic 3 when α is greater than twice the value of the maximum traffic demand. In addition, we found that the overall cost does not change for different node sequences. Considering the cooperative behavior between leaders in

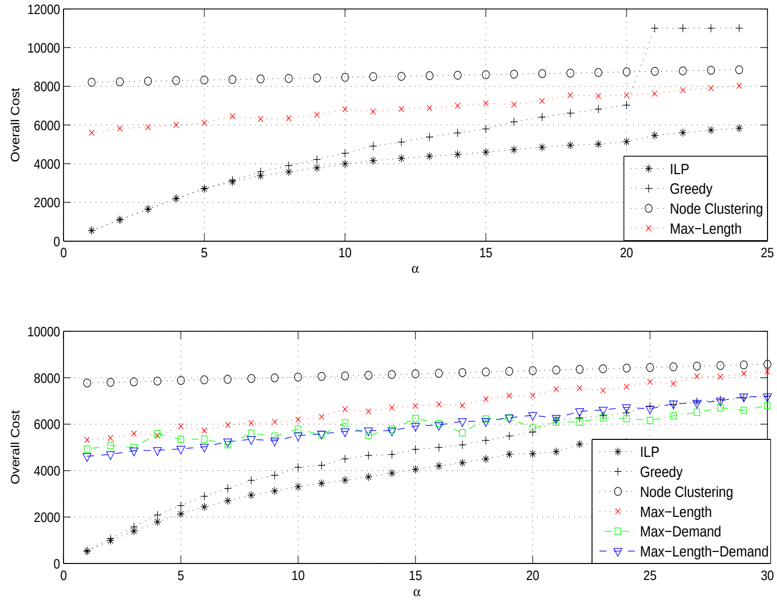

Figure 2.21: *Comparison between the different heuristics and the ILP results: a)Homogeneous traffic demand=10 b)Random traffic demand with maximum value=20*

heuristic 4, the relationship between the overall network cost and α is linear.

The order of the nodes in the node sequence defined in heuristic 3 has no effect on the results. Figure 2.22 shows the total cost function over a range of α for different node order in the node sequence.

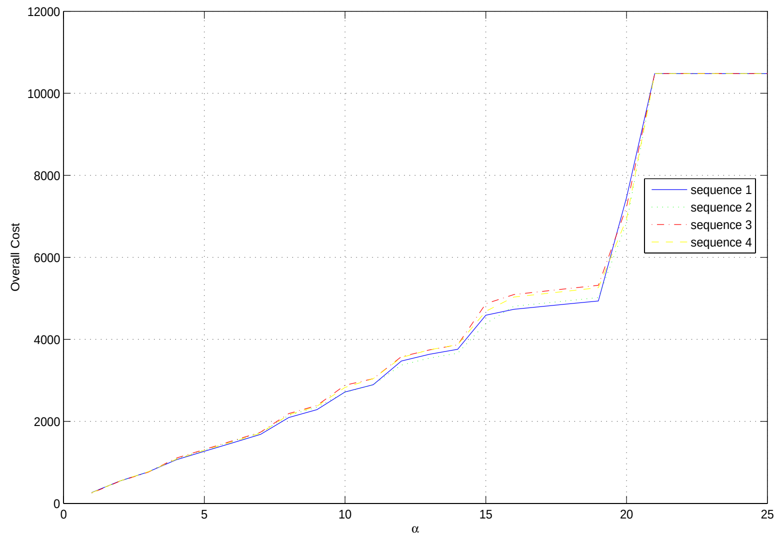

Figure 2.22: *Different node sequences and their corresponding cost function for heuristic 3*

Chapter 3

Adaptation of Overlay Network Topology

3.1 Introduction

In this chapter, we study the adptation of overlay network topologies given the traffic measurements over each link in the physical network. The Original-Destination pairs (OD pairs) in the network is estimated using the statistical signal processing tools. Based on the estimated traffic from the observed data, we can keep the overlay network cost at low levels all the time by dynamically changing the overlay topology. A simple heuristic to create the overlay topology is proposed considering the cooperative behavior of the nodes in the network. Extensive simulations are performed to observe the change in the overlay network topology at each measurement instant. The heuristic results show that the cost function of both the optimal overlay topology and the near optimal overlay topology are similar. The chapter flow starts with discussing the related work in section 3.2. The prediction tool is studied in section 3.3. In section 3.4, a heuristic is proposed to find the near optimal overlay topology. Results and some discussions are mentioned in section 3.5.

3.2 Related Work

The problem of constructing the overlay network has been addressed in [7]. The authors did not consider the OD flows pairs in the network. Recent works have shown that the OD flows affect the creation of the overlay network topology. In [25] and [14] the authors addressed the problem of creating the overlay topology taking into consideration the traffic demands between the nodes. They considered greedy and popular nodes in the network and they showed that the topology changes as the traffic demands change.

Static flows between nodes and assumed the non-cooperative and cooperative behaviors of nodes in the network given a static traffic demands to create the overlay topology was considered in [9]. In [12], the authors considered the change in the overlay topology according to some policies which depend on the change of the traffic demands.

Estimating traffic matrix has been proposed in [26], [27], [28], [29] and [8] to detect network faults, to predict future traffic volumes and to detect abnormal traffic volumes . In [8], the authors tracked the traffic volumes between OD pairs in the underlay network and predicted future traffic volume. They also considered the anomaly detection as an objective of their work. They used the well-known prediction tool Kalman filter, which has been successfully approved in tracking the traffic matrix.

3.3 Prediction Tool

Kalman Filter approach is used to predict and estimate the traffic demands between the OD flows pairs. There are N^2 different flows in the network where N is the total number of nodes in the network (The self node flow is equal to zero).

The system equations are,

$$Y_t = A_t X_t + V_t \tag{3.1}$$

$$X_{t+1} = C_t X_t + W_t \tag{3.2}$$

Where, Y_t is the vector of the collected observation. The observations are the number of packets collected every five minutes on each link at time t. The dimension of the vector Y_t is $2E \times 1$ where E is the number of physical links. A_t is the routing matrix with $a_{i,j}$ elements. $a_{i,j} = 1$ if flow j is routed over link i. The dimension of the matrix A_t is $2E \times N^2$. X_t is the traffic flow vector to be predicted with size $N^2 \times 1$. C_t is the state matrix, which represents the correlation between the different flows in the network. It also captures the progress of each flow with the time

through its diagonal elements. The size of the C_t is $N^2 \times N^2$. Both W_t and V_t denote the stochastic measurement error and the noise representing the randomness of the traffic flow respectively.
At every time instant t and using the current data observation, the traffic demands between nodes are predicted. The Kalman gain factor can be continuously adjusted according to past errors.

3.4 Approach: Greedy Heuristic

A greedy heuristic is proposed to create a near optimal overlay topology in case of the cooperative behavior of the nodes. The strategy of each node in the network is to compute the minimum of $(0.5\alpha h_{i,j}, t_{i,j}d_{i,j})$ where i and j are the source the destination nodes respectively. If $(0.5\alpha h_{i,j}$ is the minimum, node i create an overlay link with the destination node j, otherwise, the node routes the traffic demand on the physical topology. The following algorithm shows the procedure

Algorithm 6 Greedy Heuristic

```
node i: source node
node j: destination node
for i = 1 to N do
    for j = 1 to N do
        if i ≠ j then
            cost1 = 0.5αh_{i,j}
            cost2 = t_{i,j}d_{i,j}
            if cost1 ≤ cost2 then
                adjacent_{i,j} = 1
            end if
        end if
    end for
    Update the shortest path
end for
```

to implement Kalman Filter predictor including the greedy heuristic. The group of equations for each step could be found in [8]. The noise components of both V_t and W_t are assumed to have Gaussian distribution with zero mean and variances R_t and Q_t. We can drop the time subscript from A_t, C_t, R_t and Q_t. We assume that the routing scheme A_t does not change with the time. To get C_t, we assumed that all links in the physical network have enough capacities and that traffic

Algorithm 7 Kalman Filter predictor including Greedy Heuristic

Prediction step
Minimum Prediction MSE
Computing Kalman Gain
Running the Greedy Heuristic
Minimum MSE

congestion does not usually occurred due to a certain large traffic flow which could affect other flows routed on the same links. Due to the lack of NetFlow data which captures the exact traffic demand flows in the network to compute C_t, we assume different synthetic traffic demands vectors with Gaussian distribution.

3.5 Results and Discussions

Data observations were collected from [30] for 8-node network with 10 links. The observations represent the number of packets collected on each link every 5 minutes over 24 hours. For different values of α, we collected the created overlay topologies for different time instants and monitored the changes in the traffic matrices. For $1 \leq \alpha \leq 4$ the topology is the complete network for all the time instants.

For $5 \leq \alpha \leq 9$ OD pairs with 2 hops in the shortest path have high frequency of adding or dropping overlay links connecting them. Figures 3.1 and 3.2 show how the decreased traffic demands $d_{3,4}$ and $d_{4,3}$ drops the overlay link between nodes 3 and 4. For $\alpha = 10$, OD pairs with long shortest path are highly probable to add and drop links between them. Figures 3.3 and 3.4 show how the overlay link between nodes 1 and 8 is created due to the increasing in the traffic demands $d_{3,4}$ and $d_{4,3}$. For $\alpha = 15$, OD pairs with 2-hops apart have high frequency to add and drop overlay links. For $\alpha = 20$, OD pairs with 3-hops in the shortest path have high frequency to add and drop overlay links. For $\alpha = 25$, OD pairs with 3-hops apart have high frequency to add and drop overlay links. In most cases, when both flows of the OD node pairs change simultaneously, the topology is changed regardless the value of α. When the routing scheme of the traffic demands between far nodes changes, congestion at the routers has a high chance to

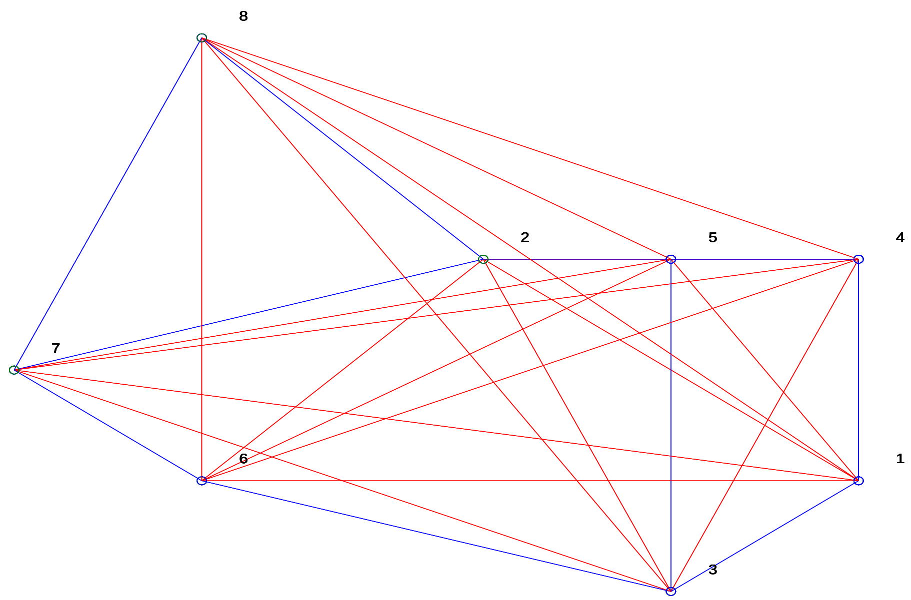

Figure 3.1: *Created overlay topology with* $\alpha = 5$ *at time instant=4 with traffic demand* $d_{3,4} = 3$ *and* $d_{4,3} = 9$

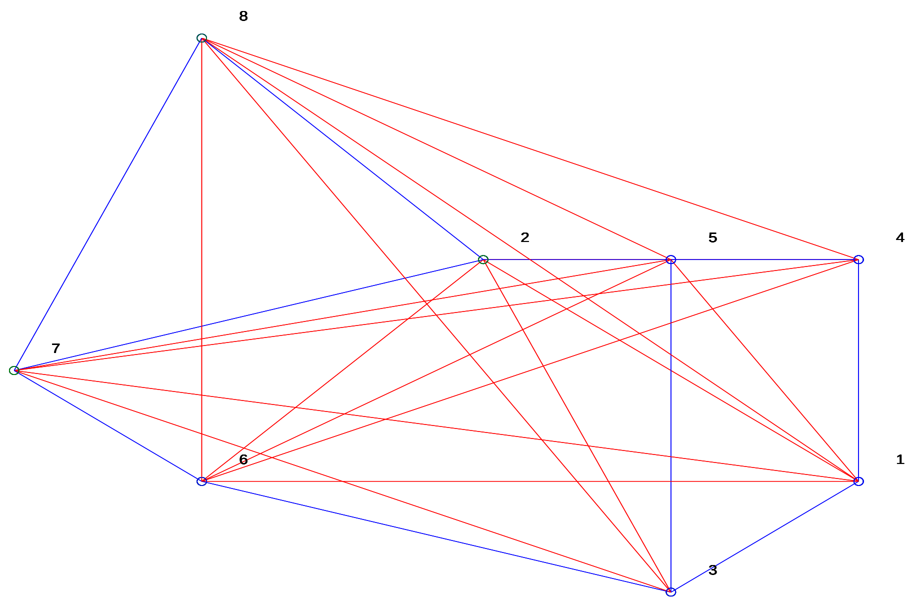

Figure 3.2: *Created overlay topology with* $\alpha = 5$ *at time instant=5 with traffic demand* $d_{3,4} = 2$ *and* $d_{4,3} = 8$

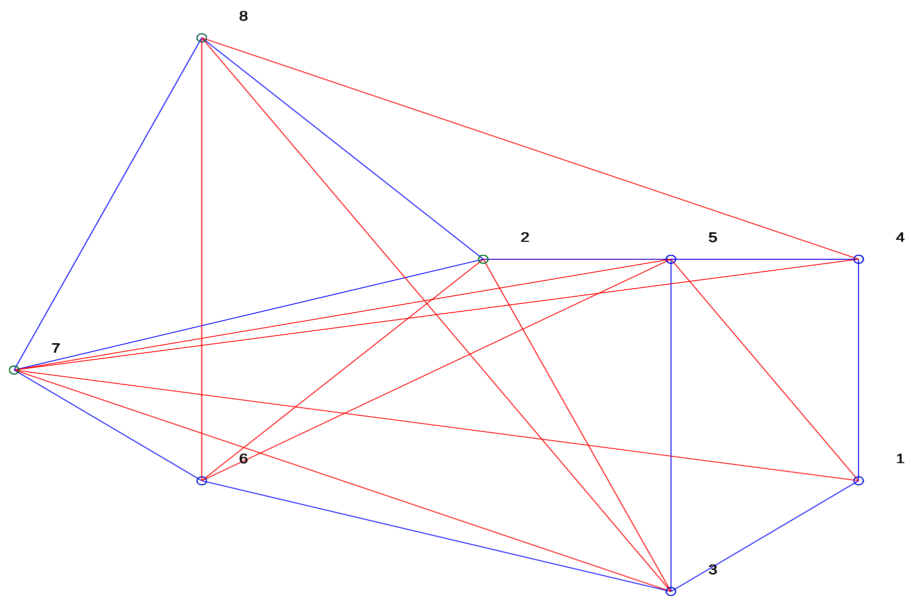

Figure 3.3: *Created overlay topology with* $\alpha = 10$ *at time instant=5 with traffic demand* $d_{1,8} = 4$ *and* $d_{8,1} = 15$

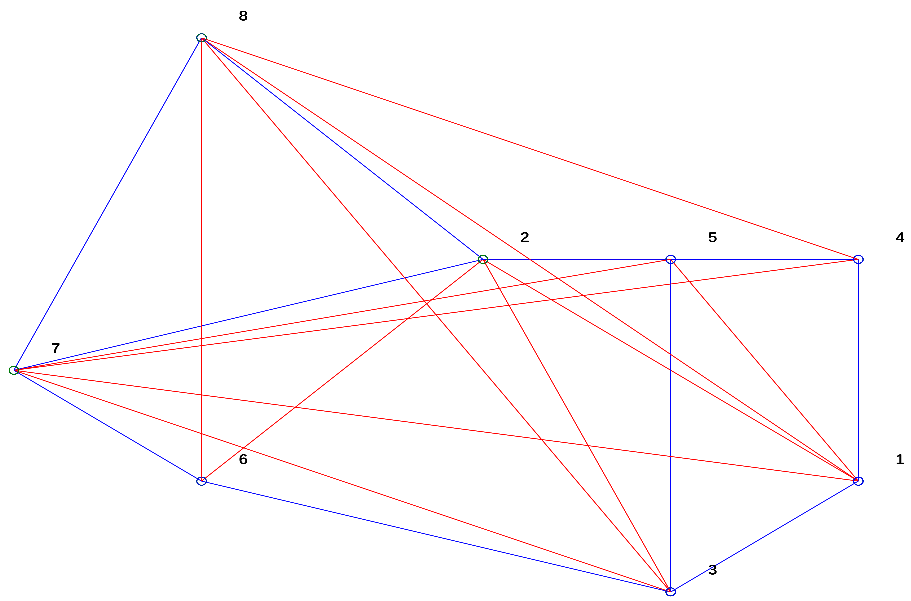

Figure 3.4: *Created overlay topology with* $\alpha = 10$ *at time instant=6 with traffic demand* $d_{1,8} = 5$ *and* $d_{8,1} = 15$

occur. Besides, it may disrupt the other OD flows in the network.

Chapter 4

Design of Hybrid Optical Network Topology for Supporting Multicast

4.1 Introduction

Optical networks offer tremendous bandwidth to transfer information between different network sites. Bandwidth in the optical networks is a major network resource that has to be maximally utilized. Bandwidth over each optical link is divided to channels which individually still represents huge bandwidth. Then a channel is subdivided to subchannels allowing a certain degree of sharing. The division of the bandwidth allows the creation of virtual links and virtual topologies. A light-path is a logical channel connection between two different optical nodes. A light-tree is the general case of the light-path given multipoint of data transfer. Each light-path and light-tree is implemented on a single channel on a physical link.

Over each channel, different network applications are running which includes multicast traffic. A multicast session is a point to multipoint data transfer for an application. Video on demand, webcast channels and online applications are examples of multicast traffic recently used by network users. All nodes in the multicast session receive the same copy of the multicast packets via network duplication rather than multiple unicast. Multicast sessions are supported using a hybrid combination of light-paths, light-trees and the available channels on physical links. The hybrid topology exploits the channels to increase number of supported multicast sessions. The problem of supporting multicast sessions given the physical links and the light-paths is formulated using the Integer Linear Programming. The degree of sharing the channels are given for both physical links and light-paths. Then, we consider that the light-trees are given, then we implement the light-paths and light-trees on the physical link channels using the ILP. This formulation compute the available channels on the physical links. Finally, we formulate the problem of creating the hybrid topology over WDM network to support different multicast sessions using the ILP. The optimal hybrid topologies generated from the ILP formulation are analysed to observe the preference of using light-paths, light-trees and available channels on the physical links.

The organization of this chapter starts with the related work in section 4.2. The problem formulation is explained in section 4.3. Results and analysis are discussed in section 4.4.

4.2 Related Work

Several studies have been done on design the virtual topology. A group of light-paths compose the virtual topology. Recent work studied the design of light-paths to minimize the optics-electronic-optics O/E/O conversion while routing the packets in the network. Each light-path starts with and E/O conversion devices. The light-path includes several physical links. The termination of the light-path is O/E conversion devices. The light-path concept was proposed to minimize the cost of using the electronic-optic transceivers in the network.

In [31], the authors introduce the light-tree concept. Light-trees were proposed to enhance the performance of the WDM routed networks. the authors also proposed optimization program to formulate the problem of finding optimum virtual topology. The objective of the formulation is to minimize the average packet hop distance and the total number of required transceivers in the network.

In [32], the authors introduced the optical transport network OTN. The OTN was proposed for need of data for bandwidth and the emergence of new broadband services. The authors also proposed the service optical network SON which allows building on the OTN infrastructure to provide service management and switched connections.

The authors in [10] formulated the problem of creating hybrid topology for multicast session over constrained WDM networks using ILP. The network resources are the virtual topology representing the light-paths and the physical topology representing light-trees. They proposed the degree of sharing over the physical links and light-paths. In our work, we assume the existence of some light-trees designed over individual channels on the physical links. The degree of sharing is proposed for light-paths and light-trees so that more multicast sessions are supported.

4.3 Problem Formulation

The problem of generating hybrid topology to support multicast sessions is formulated using the Integer Linear Programming. We introduce several ILP formulations which are used to construct the hybrid topology depending on the given data input. At the beginning, we only consider the physical topology which consists of the physical links and the virtual topology which consists of the light-paths. Then, we suppose that the light-trees are given in the data input. The light-trees and light-paths are implemented on the physical topology. Since each light-path and light-tree uses one channel to be implemented, we compute the available (remaining) channels on the physical links using ILP. The last ILP formulation represents the main problem where the available channels on the physical links, the light-tree topologies and the degree of sharing each light-tree and light-path are given as input data to the formulation.

4.3.1 Creation of hybrid topologies given the physical and virtual topologies

We formulate the problem given the physical links and the light-paths. First, we describe the data input to the problem and the decision variables used to formulate the problem. Second, we discuss the formulation of the objective function and explain the set of constrain equations.

Data Input

- Physical Network $padjacent_{m,n}$: It is the adjacency matrix of N-node physical network. Each element has the value either 1 or 0 depending on the presence of a physical link between nodes m and n.

- Virtual Topology $vadjacent_{m,n}$: It is the adjacency matrix of N-node virtual network. Each element has the value either 1 or 0 depending on the presence of a light-path between nodes m and n.
- Multicast Sessions $session_{i,k}$: It is a binary data representing the member destination nodes k in the multicast session i.
- Source nodes $source_i$: It contains the source node of each multicast session.
- Weight of physical links $w_{m,n}$: It is the weighting cost assigned to every physical link in the network.
- Weight of virtual Links α: It is a homogeneous weighting cost of using a light-path in the network.
- Maximum number of wavelengths on each physical and virtual link Cp and Cv.

Decision Variables

- $M_{i,m,n}$: Binary variable which is equal to 1 if the physical link m, n is used for the multicast session i.
- $Y_{i,m,n}$: Binary variable which is equal to 1 if the light-path m, n is used for the multicast session i.
- $f_{i,m,n}$: Integer variable which represents the flow accommodation from the source node in the session 1 over the different physical links m, n.
- $y_{i,m,n}$: Integer variable which represents the flow accommodation from the source node in the session 1 over the different light-paths m, n.

Objective function

$$minimize \sum_i \sum_m \sum_n (w_{m,n} M_{i,m,n} + \alpha Y_{i,m,,n}) \tag{4.1}$$

The objective function is to minimize the total weight cost of using light-paths and physical links. The goal of the objective function is to select low cost physical links and the shortest path for the light-paths since the cost of selecting a light-path is homogeneous.

Constraint Equations

$$\sum_{n}(y_{i,source_i,n} + f_{i,source_i,n}) = \sum_{k} session_{i,k}\ \forall\ i \tag{4.2}$$

$$\sum_{m}(y_{i,m,n} - y_{i,n,m} + f_{i,m,n} - f_{i,n,m}) = session_{i,n}\ \forall\ i, n, n \neq source_i \tag{4.3}$$

$$y_{i,m,source_i} + f_{i,m,source_i} = 0\ \forall\ i, m \tag{4.4}$$

$$f_{i,m,n} \leq M_{i,m,n} \sum_{k} session_{i,k}\ \forall i, m, n \tag{4.5}$$

$$\sum_{i} M_{i,m,,n} \leq Cp\ \forall m, n \tag{4.6}$$

$$y_{i,m,n} \leq Y_{i,m,n} \sum_{k} session_{i,k}\ \forall i, m, n \tag{4.7}$$

$$\sum_{i} Y_{i,m,,n} \leq Cv\ \forall m, n \tag{4.8}$$

In the formulation, Eq. 4.2 means that the source node of each multicast session has to send all the traffic to its destination nodes. Eq. 4.3 represents the flow conservation equation over the selected physical links and light-paths. The source node does not receive a packet from the same multicast session as shown in Eq. 4.4. Eq.4.5 constrains the traffic to flow on the selected physical link for a given multicast session. Eq.4.6 constrains number of multicast sessions running over the same physical link to the maximum number of wavelength channels Cp. Eq.4.7 and Eq.4.8 are similar to Eq.4.5 and Eq.4.6 respectively in case of the light-paths.

This formulation is equivalent to the formulation given in [10] in the functionality but with reduced complexity. We combine both the light-tree construction constraints and the flow conservation constraints into one group of constraints to decrease number of constraint equations and to improve the running time of the formulation.

4.3.2 Implementation of the light-paths and light-trees on the physical topology

In this formulation, light-paths and light-trees are implemented over the physical links given number of channels. The available channels are computed to be input data for the main ILP formulation.

Data Input

The data inputs are the physical topology $padjacent_{m,n}$, the virtual topology $vadjacent_{m,n}$, the light-tree member $treeMember_{i,m}$ which is boolean data representing the node members of light-tree i, $startNode_i$ which are the nodes from which the light-trees are constructed and the maximum capacity (total number of channels) on each physical link Cp.

Decision Variable

The decision variables are $lightpathFlow_{s,m,n}$ which is a flow variable representing the flow of a light-path on link m, n for the node s, $treeFlow_{i,m,n}$ which is a flow variable representing the flow of a light-tree, $tree_{i,m,n}$ which is the topology of the constructed light-tree i, $usedChannels_{m,n}$ representing number of used channels on each physical link and $Cpavailable_{m,n}$ which is the number of available channels on the physical link m, n.

Objective Function

$$minimize \sum_s \sum_m \sum_n lightpathFlow_{s,m,n} + \sum_i \sum_m \sum_n treeFlow_{i,m,n} \tag{4.9}$$

The objective function is minimizing the flow variables which are used to get the implementation of both light-paths and light-trees on the physical link.

Constraint Equations

Light-path

$$\sum_n ligthpathFlow_{s,s,n} = \sum_k vadjacent_{s,k} \ \ \forall s \tag{4.10}$$

$$\sum_m (lightpathFlow_{s,m,n} - lightpathFlow_{s,n,m}) \leq vadjacent_{s,n} \ \ \forall s \ n \neq s \tag{4.11}$$

$$lightpathFlow_{s,m,n} \leq padjacent_{m,n}\, C \ \ \forall s \ m \ n \tag{4.12}$$

$$\sum_s lightpathFlow_{s,m,n} \leq padjacent_{m,n}\, C \ \ \forall m \ n \tag{4.13}$$

$$lightpathFlow_{s,m,s} + lightpathFlow_{s,m,m} = 0 \tag{4.14}$$

This formulation is equivalent to the formulation of computing the shortest path between any pair of node in case of the light-path. Eq. 4.10 represents the virtual topology as a traffic demand matrix of one unit between each node pair with a light-path connecting them. Eq. 4.11 is the flow conservation equation for the light-path. Eq. 4.12 means that if there is a physical link, flow could be allocated over it. C is a big number. Eq. 4.13 allows any node to allocate the traffic over the existing physical link. Eq. 4.14 ensures that a source node will not receive a flow over that light-path and a node does not have a self traffic.

Light-tree

$$\sum_n treeFlow_{i,startNode_i,n} = \sum_m treeMember_{i,m} - 1 \, \forall \, i \tag{4.15}$$

$$\sum_{m}(treeFlow_{i,m,n} - treeFlow_{i,n,m}) = treeMember_{i,n} \; \forall\; i, n \neq startNode_i \tag{4.16}$$

$$treeFlow_{i,m,n} \leq padjacent_{m,n}\; C \; \forall\; i, m, n \tag{4.17}$$

$$treeFlow_{i,m,n} \leq tree_{i,m,n}\; C \; \forall i, m, n \tag{4.18}$$

$$tree_{i,m,n} = tree_{i,n,m} \; \forall i, m, n \tag{4.19}$$

$$\sum_{s} lightpathFlow_{s,m,n} + \sum_{i} tree_{i,m,n} = usedChannels_{m,n} \; \forall m, n \tag{4.20}$$

$$padjacent_{m,n} Cp - usedChannels_{m,n} = Cpavailable_{m,n} \; \forall m, n \tag{4.21}$$

This formulation is equivalent to the formulation of constructing a minimum spanning tree MST in case of the light-path. Eq. 4.15 computes the total flow sent by the $startNode_i$ towards each tree member. Eq. 4.16 represents the flow conservation over the light-tree. The tree flow can use and physical link as shown in Eq. 4.17. Eq. 4.18 constructs the light-tree topologies $tree_{i,m,n}$ based on the links used by the light-tree flows. Eq. 4.19 makes the light-tree topology matrices to be symmetric. Eq. 4.20 computes the total used capacity on each physical link given the implementation of light-trees and light-paths flows on the physical links. Eq. 4.21 computes the available capacity on each physical link $Cpavailable_{m,n}$ given the number of used channels on each physical link and the total capacity Cp over each link.

4.3.3 Creation of hybrid topology given virtual, physical and light-tree topologies

This formulation represents the main problem of creating hybrid optical topology to support multicast sessions. virtual, physical and light-tree topologies are given as input data. The network channel resources are the available channels on the physical links, the degree of sharing the light-paths Cv and the degree of sharing the light-trees Ct among different multicast sessions.

Data Input

- $padjacent_{m,n}$: The adjacency matrix of the physical topology. $padjacent_{m,n} = 1$ if there is a link between m and n nodes.
- $vadjacent_{m,n}$: The adjacency matrix of the virtual topology. $vadjacent_{m,n} = 1$ if there is a link between m and n nodes.
- $tree_{i,m,n}$: The symmetric adjacency matrix of the light-tree topology i.
- $source_s$: Source nodes of the multicast sessions.
- $session_{s,m}$: It is a binary data representing the node members of the multicast sessions. $session_{s,m} = 1$ if for session s the node m is a member.
- w: Link weight of the physical topology.
- $\alpha_{m,n}$: Link weight of the virtual topology.It is a function of number of physical links supporting each light-path.
- β: It is the weight cost of using the light-trees.
- $Cpavailable_{m,n}$: number of available channels on each physical link.
- Cv: number of available sharing channels on each virtual link.
- Ct: number of available sharing channels on the light-trees.

Decision Variables

- $M_{s,m,n}$: Binary variable which is equal to 1 if the physical link m, n is used for the multicast session s.

- $Y_{s,m,n}$: Binary variable which is equal to 1 if the light-path m, n is used for the multicast session s.

- $T_{s,i,m,n}$: Binary variable which is equal to 1 if the physical link m, n in the corresponding chosen light-tree i is used for the multicast session s.

- $\nu_{s,i}$: It is a binary variable. $\nu_{s,i} = 1$ if the light-tree i is chosen to satisfy the session s.

- $f_{s,m,n}$: Integer variable which represents the flow accommodation from the source node in the session s over the different physical links m, n.

- $y_{s,m,n}$: Integer variable which represents the flow accommodation from the source node in the session s over the different virtual links m, n.

- $t_{s,i,m,n}$: Integer variable which represents the flow accommodation over the selected light-tree i to satisfy the multicast session s.

- $member_{s,m}$: $member_{s,m} = 1$ if for the session s, the node m is either a member of a light-tree topology, a destination of the session or an intermediate node.

Objective function

$$minimize \sum_s \sum_m \sum_n (wM_{s,m,n} + \alpha_{m,n} Y_{s,m,n} + \beta \sum_i T_{s,i,m,n}) \tag{4.22}$$

The objective function represents the cost of the selected hybrid topology components. The cost of selecting the physical links reflects the path with minimum hops while w is homogeneous. The cost of selecting a light-path is $\alpha_{m,,n}$ which is equal to number of physical links used to design that light-path. The cost of a light-tree is β. The objective function indirectly minimizes number of non-destination intermediate nodes between different physical links and light-paths. It also selects light-trees which have minimum number of fortuitous nodes [33]. (A node, in the light-tree, is a fortuitous destination when it receives a copy of a multicast session packets and it is not a member in that session [34]).

Constraints

Selecting Light-trees:

The following group of equations choose light-tree (light-trees) i to satisfy the multicast session s. They ensure that the variable $T_{s,i,m,n}$ of constructing the selected light-tree will not mess up the topology of the selected light-tree. Besides, these equations will ensure that the light-tree flow $t_{s,i,m,n}$ will be in the correct direction.

$$T_{s,i,m,n} \leq tree_{i,m,n} \, \forall \; s, i, m, n \tag{4.23}$$

$$T_{s,i,m,n} + T_{s,i,k,n} \leq 1 \;\; \forall \; s, i, m, n, k, k \neq m \tag{4.24}$$

$$\sum_{m} \sum_{n} T_{s,i,m,n} = \sum_{m} \sum_{n} \nu_{s,i} tree_{i,m,n} \;\; \forall s, i \tag{4.25}$$

$$T_{s,i,m,n} + T_{s,i,n,m} + M_{s,m,n} + M_{s,n,m} + Y_{s,m,n} + Y_{s,n,m} \leq 1 \, \forall \; s, i, m, n \tag{4.26}$$

$$\sum_{i} \sum_{n} (T_{s,i,m,n} + T_{s,i,n,m}) + session_{s,m} \leq member_{s,m} \, C \;\; \forall m, s \tag{4.27}$$

Eq. 4.23 means that if a link m, n is part of a light-tree i, it is used to satisfy the multicast demand s. Eq.4.24 avoids the situation that a certain node in the light-tree i receives more than a packet for the same multicast session s. Eq. 4.25 with Eq. 4.23 guarrentees that the variable $T_{s,i,m,n}$ will include all the links of the selected light-tree i to support the multicast session s. Eq. 4.26 avoids the situation that different resources (channels) could be used more than one time for the same session s on the same link m, n (or light-path m, n). It also avoids the situation where the traffic can flow on the same link m, n in different directions (m, n and n, m) for the same session s. Eq. 4.27 assigns the members of the selected light-trees to the total members of the solution beside the members of the original multicast session s. C is a big number.

Traffic Flow Equations:

$$\sum_{n}(y_{s,source_s,n} + f_{s,source_s,n} + \sum_{i} t_{s,i,source_s,n}) = \sum_{k} members_{s,k} - 1 \;\; \forall s \; n \neq source_s \tag{4.28}$$

$$\sum_{m}(y_{s,m,n} - y_{s,n,m} + f_{s,m,n} - f_{s,n,m} + \sum_{i}(t_{s,m,n} - t_{s,n,m})) = members_{s,n} \;\; \forall s \; n \neq source_s \tag{4.29}$$

$$Y_{s,m,source_s} + M_{s,m,source_s} + T_{s,i,m,source_s} = 0 \;\; \forall \; s, i, m \tag{4.30}$$

$$t_{s,i,m,source_s} = 0 \;\; \forall \; s, i, m \tag{4.31}$$

In Eq. 4.28, the source node of each session $source_s$ sends the traffic demand to all the destination using the virtual and physical links and the light-trees attached with the source node. The destinations are the multicast session s members, intermediate nodes and the light-tree members if an existing light-tree is selected to satisfy the demand for the session s. Eq.4.29 represents the flow conservation equation. Eq.4.30 ensures that the source node of session s will not receive a multicast packet from the same session. The source node has no traffic demand for each multicast session for the light-tree flow as shown in Eq.4.31.

Channel Constraints:

$$f_{s,m,n} \leq M_{s,m,n} \, C \;\; \forall s, m, n \tag{4.32}$$

$$\sum_{s} M_{s,m,,n} \leq Cpavailable_{m,n} \;\; \forall m, n \tag{4.33}$$

$$y_{s,m,n} \leq Y_{s,m,n} \, C \;\; \forall s, m, n \tag{4.34}$$

$$\sum_{s} Y_{s,m,,n} \leq Cv \;\; \forall m, n \tag{4.35}$$

$$t_{s,i,m,n} \leq T_{s,i,m,n}\, C \;\; \forall s,i,m,n \tag{4.36}$$

$$t_{s,i,m,n} \geq T_{s,i,m,n} \;\; \forall s,i,m,n \tag{4.37}$$

$$\sum_{s} \nu_{s,i} \leq Ct \;\; \forall i \tag{4.38}$$

$$M_{s,m,n} \leq padjacent_{m,n} \;\; \forall s,m,n \tag{4.39}$$

$$Y_{s,m,n} \leq vadjacent_{m,n} \;\; \forall s,m,n \tag{4.40}$$

$$\sum_{n} (Y_{s,m,n} + M_{s,m,n}) \leq member_{s,m}\, C \;\; \forall s,m \tag{4.41}$$

Eqs. 4.32-4.35 have the same functions as Eqs. 4.5-4.8 except replacing Cp with $Cpavailable_{m,n}$ and including C which is a big number. Eqs.4.36 and 4.37 force the traffic to flow on the links m,n of the selected light-tree i and to be in the proper direction. Eq. 4.38 constrains each light-tree i to be used up to Ct times where Ct is the degree of sharing light-tree i among different multicast sessions. Eqs. 4.39 and 4.40 constrain the selection of a physical link and a light-path between the existing ones.Eq. 4.41 guarantees that all the intermediate nodes of the selected physical links and light-paths are included in $member_{s,m}$ for the minimization purpose.

4.4 Results and Evaluation

Extensive simulations have been conducted to investigate the validation, feasibility and efficiency of the ILP formulation. The network used in evaluation is 7-node network. Figure 4.1 shows the physical and virtual topologies while figure 4.2 shows the different used light-trees. Each link carries a limited channels. We assumed that number of channels on each physical link used for the

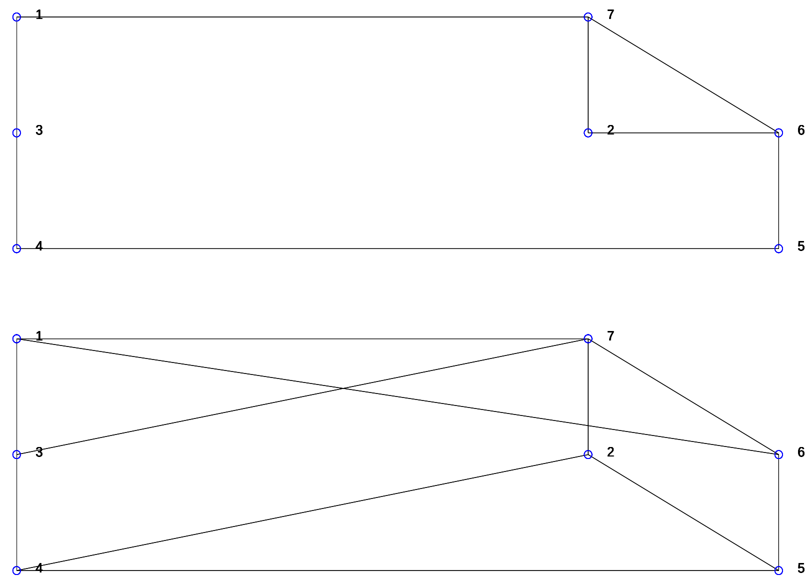

Figure 4.1: *Physical and virtual Topologies (from top to bottom)*

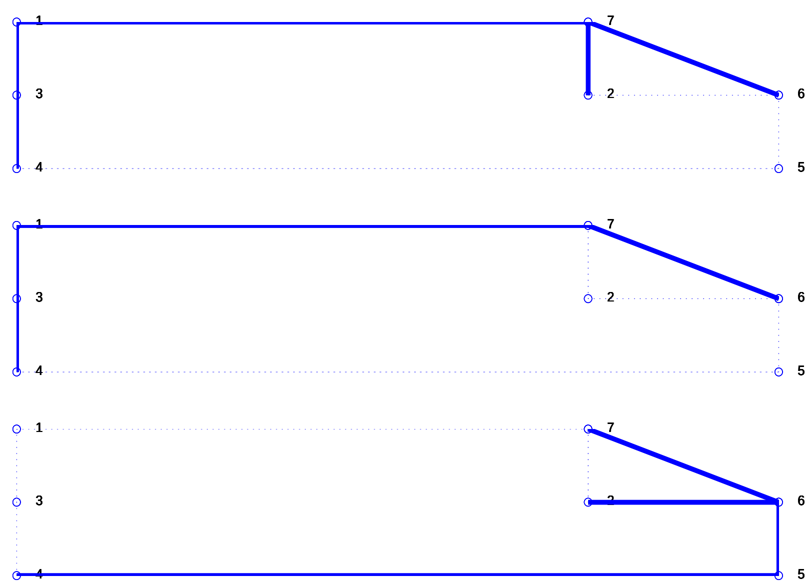

Figure 4.2: *From up to down: Tree 1, Tree 2 and tree 3 Topologies*

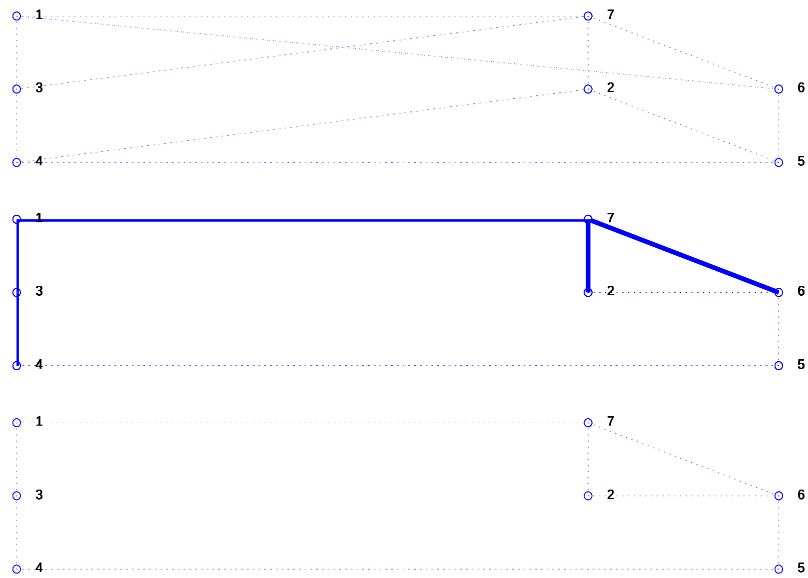

Figure 4.3: *From up to down: virtual, light-tree and physical topologies for session 1. Solid lines: used links, dotted lines: available links*

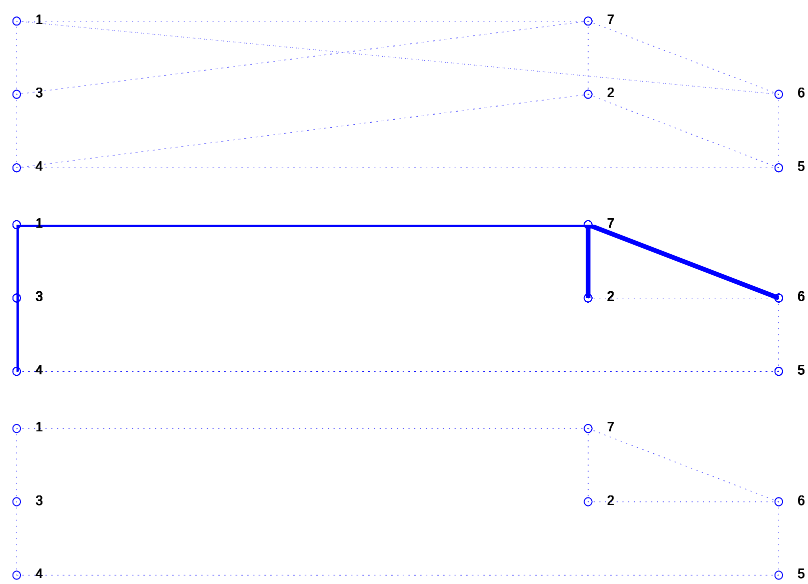

Figure 4.4: *From up to down: virtual, light-tree and physical topologies for session 2. Solid lines: used links, dotted lines: available links*

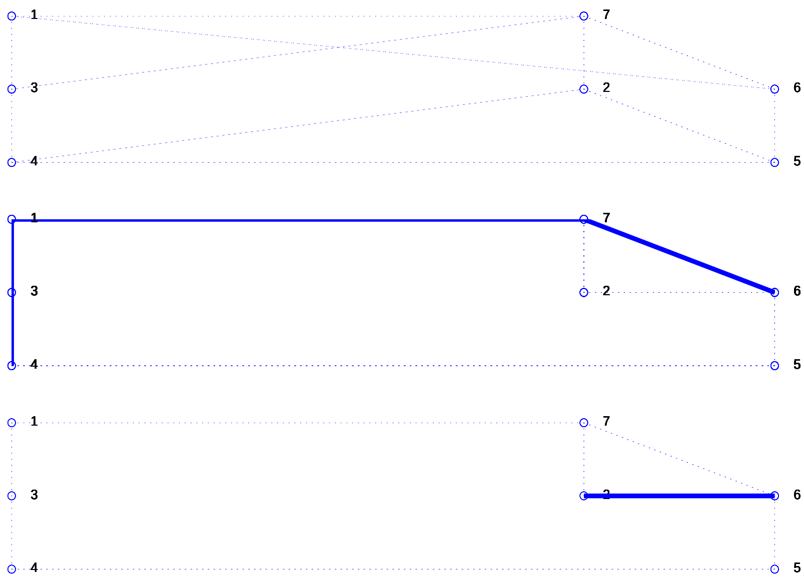

Figure 4.5: *From up to down: virtual, light-tree and physical topologies for session 3. Solid lines: used links, dotted lines: available links*

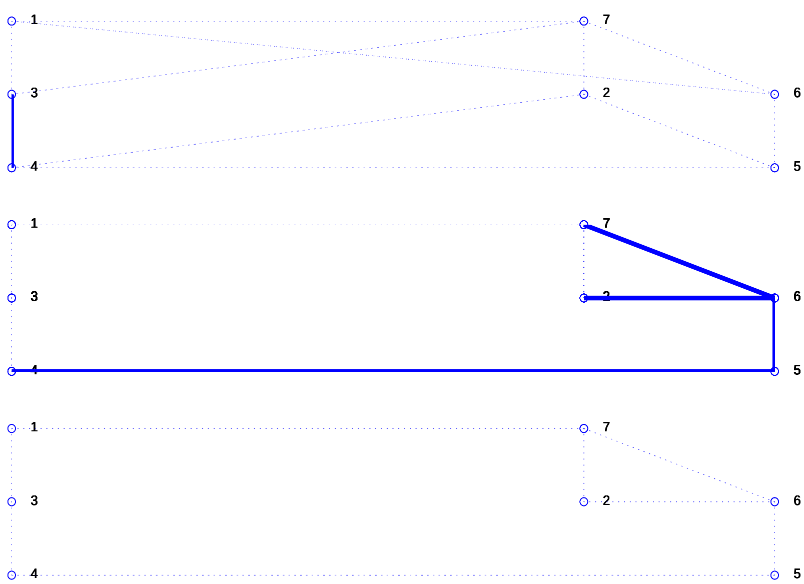

Figure 4.6: *From up to down: virtual, light-tree and physical topologies for session 4. Solid lines: used links, dotted lines: available links*

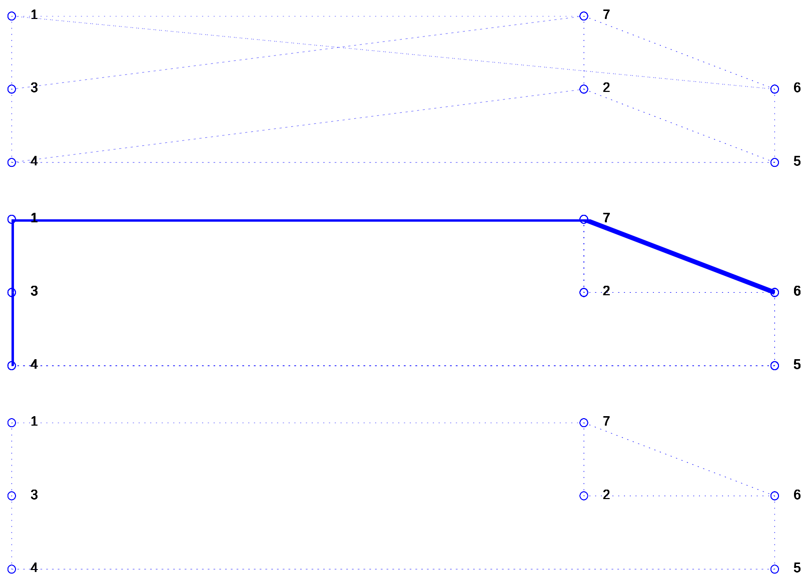

Figure 4.7: *From up to down: virtual, light-tree and physical topologies for session 5. Solid lines: used links, dotted lines: available links*

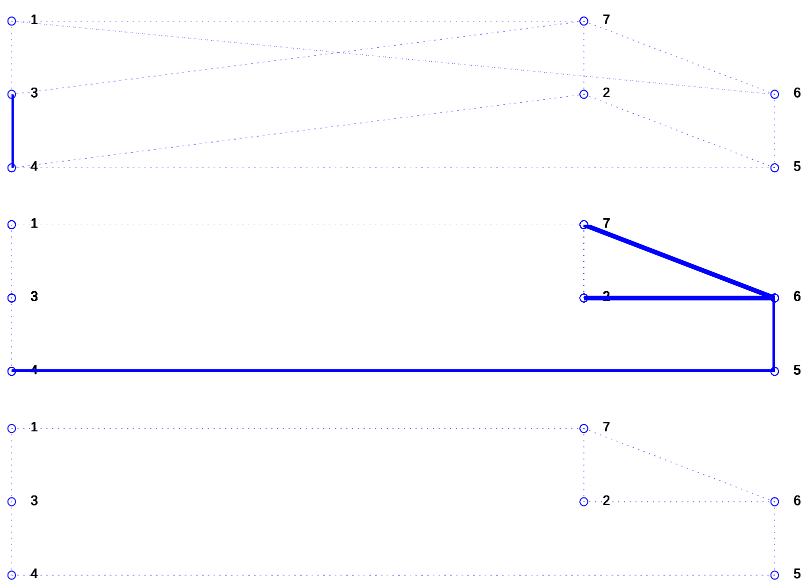

Figure 4.8: *From up to down: virtual, light-tree and physical topologies for session 6. Solid lines: used links, dotted lines: available links*

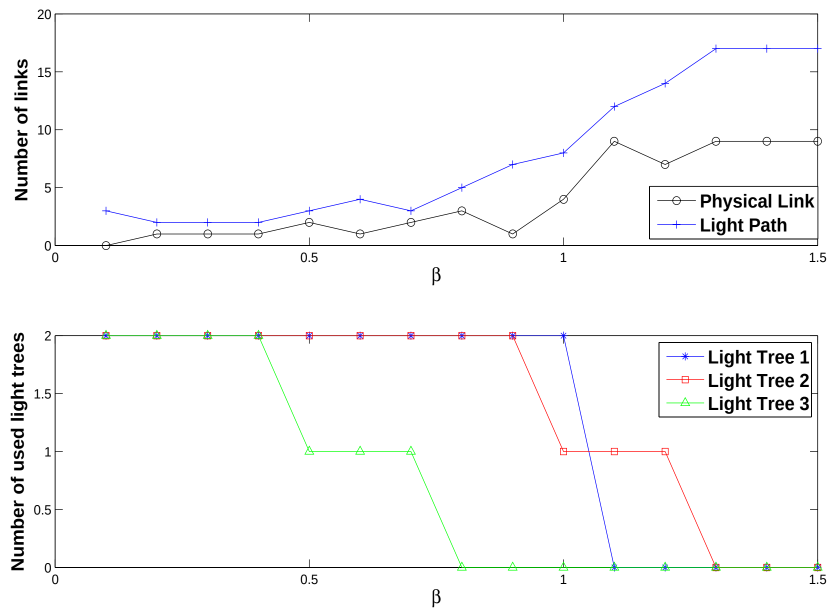

Figure 4.9: *From up to down: Number of links and light-trees vs. β.*

second formulation is $Cp = 5$ to implement the given light-trees and light-paths and to compute the available channels.

Six multicast sessions have been used to evaluate the formulation, S_1={1, 2, 3, 4, 6, 7}, S_2={1, 2, 3, 4, 6, 7}, S_3={1, 2, 3, 4, 7}, S_4={2, 3, 4, 6, 7}, S_5={1, 3, 4, 6, 7} and S_6={3, 4, 5, 6} with source nodes {4, 6, 1, 7, 1, 5}. Figures 4.3, 4.4, 4.5, 4.6, 4.7 and 4.8 show the created hybrid optical topologies for each session using the ILP formulation given $w = 1$, $\beta = 0.01$, $Cv = 2$ and $Ct = 2$.

Small value of β means that we increases the flavor of using the light-trees over the light-paths and physical links. In the figures representing the sessions topologies, the light-trees are exploited entirely according to the degree of sharing Ct. The light-path and the physical links are not used extensively due to their high cost coefficients w and $\alpha_{m,n}$ with respect to the cost coefficient of the light-tree β. If we decrease number of multicast sessions and the cost coefficient of using the light-trees is small, each session can use more than a light-tree to satisfy its demand.

Figures 4.9 shows the variation of the topologies when β is changed and w is constant and is

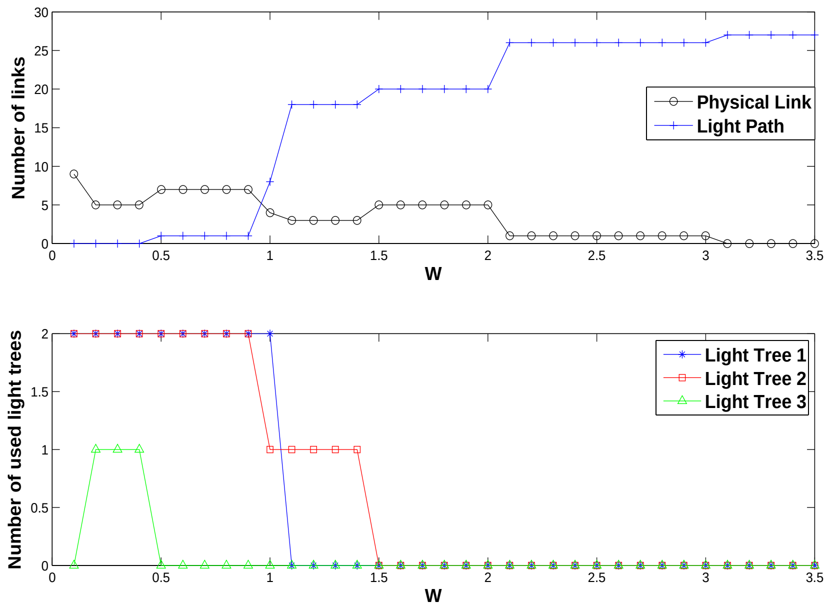

Figure 4.10: *From up to down: Number of links and light-trees vs.* w.

equal to 1 where $Cv = Ct = 2$. As β increases, the preference of using the light-trees decreases and number of used light-paths and physical links increase. When a light-tree is no longer used, light-paths and physical links are used to balance the required resources to satisfy the traffic demands. Similarly, figure 4.10 shows the variation of the topologies when w is changed and β is constant and is equal to 1 where $Cv = Ct = 2$. The figure shows that number of used physical links are increasing and decreasing with the change in number of used light-trees respectively. Number of used light-paths increases when w becomes more expensive to equalize the decrease in number of light-trees and physical links.

Finally, we captured number of supported multicast sessions with degree of sharing the light-paths and the light-trees. We assumed that $Cv = Ct$. The figure show that number of multicast sessions increases when degree of sharing increases. Our approach supports more multicast sessions comparing with number of supported multicast sessions in [10].

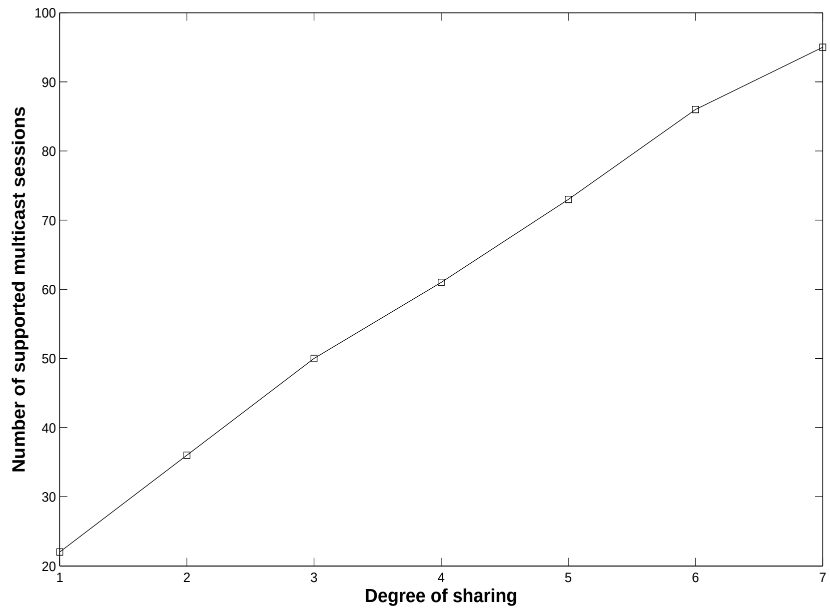

Figure 4.11: *Number of supported multicast sessions vs. degree of sharing the light-paths and the light-trees.*

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The objective of this thesis is to find the optimal topology design for different virtual networks. For the overlay network topology, we considered both the routing cost and the overlay link creation cost. We formulate the problem using the Integer Linear Programming for both the non cooperative and cooperative node behaviors. In addition, we propose some heuristics to select the near optimal topology when the problem size increases. Different static traffic scenarios are used in simulation : homogeneous traffic, uniform random traffic and bimodal random traffic demands. The networks used in the simulations are real networks with different sizes and with different topology characteristics.
Our results showed that the selection of the best heuristic among the set of the proposed ones is a function of α. The optimal and near optimal overlay topologies, generated by the ILP formulation and the heuristics respectively, are characterized to understand how nodes behave in the networks. The effect of the traffic demands and number of hops separating pair of nodes on creating the overlay topologies are shown. The effect of the underlay topologies on creating the overlay topologies is studied using some topological metrics.

We studied a new approach to adapt the overlay network topology based on estimating and predicting the traffic volume in the underlay network for dynamic traffic demands. The same cost function is used to generate the overlay topology at each time instant. The cost function includes the OD pair flows which affect on the creation of the overlay topology. The OD flows pairs are estimated using the Kalman filter. Predicting the traffic matrix will keep the overlay network topology dynamically adapted to achieve low cost when the traffic volume is changed. The results show that the complete network graph is the best solution to keep the topology unchanged regardless the change in the traffic demand. Frequently adding and dropping overlay links between nodes separated by the longest shortest paths has to be avoided.

Hybrid optical topology design over constrained WDM network is generated to support different multicast traffic demands. The problem of creating the bybrid topology is first formulated given the virtual topology and the physical topology. The virtual topology is composed by light paths while the physical topology is composed by the individual physical links. We also consider the light tree structure to enhance number of supported multicast sessions. We implement the light paths and the light trees on the physical topology given number of wavelength channels on each physical link. Each light tree and light path exploits a single channel to be implemented. The available channels are computed on the physical links. The problem of creating hybrid optical topology is formulated using ILP given the light trees, virtual and physical topologies. The degree of sharing the light tree and the light path is used such that more than a multicast session can use the resource.

Extensive simulations are performed over a small optical network with different multicast sessions. Our approach shows how the virtual, physical and light trees topologies are exploited given different degrees of cost coefficients. Number of supported multicast sessions increases with the increase of degree of sharing the light trees and the light path.

5.2 Future Work

Future work will focus on generating the optimal overlay topologies for a wide range of α by determining cutting planes that decrease the time required to solve the Integer Linear Programming formulation. Future work will also focus on studying the overlay topology creation and adaptation in case of unknown traffic demands. A hybrid cooperative behavior of nodes will be studied which merges the cooperative and the non cooperative behaviors of the nodes. Heterogeneous values of the overlay cost coefficient will be proposed for each node in the network, and its effect on the overlay topology creation will be studied.

The future work will also concentrate on estimating the traffic demands in a decentralized way. A reconfiguration cost will be proposed to adapt the overlay topology to minimize the disruption of end-to-end flows. We will also propose semi-dynamic overlay topology design. Beside, a heuristic to change the value of the overlay coefficient with the traffic demand for the overlay static topology will also be proposed.

For the hybrid optical topology design, heuristics will be proposed to create near-optimal hybrid optical topologies in polynomial time and to support many multicast sessions. Future work will also concentrate on studying the blocking probability of a multicast session based on the available channels. Larger optical network will be used to test the proposed heuristics.

Bibliography

[1] X. Gu, K. Nahrstedt, R. Chang, and C. Ward, Qos-assured service composition in managed service overlay networks, in *In Proc. IEEE 23rd International Conference on Distributed Computing Systems, Providence.*, 2003.

[2] S. Baset and H. Schulzrinne, An analysis of the skype peer-to-peer internet telephony protocol, in *In Proceedings of the INFOCOM '06*, Barcelona, Spain, April 2006.

[3] S. Vieira and J. Liebeherr, Topology design for service overlay networks with bandwidth guarantees, in *Proceedings of IWQoS 2004*, Montreal, Canada, June 2004.

[4] Z. Li and P. Mohapatra, Qron: Qos-aware routing in overlay networks, in *Selected Areas in Communications, IEEE Journal*, 2004.

[5] B. Zhao et al., Tapestry: A resilient global-scale overlay for service deployment, in *IEEE Journal on Selected Area in Communications*, *Special Issue on Service Overlay Networks*, volume 22, 2004.

[6] J. Han, D. Watson, and F. Jahanian, Topology aware overlay networks, in *Proceedings of IEEE INFOCOM'05*, Miami, USA, March 2005.

[7] B. Chun, R. Fonseca, I. Stoica, and J. Kubiatowicz, Characterizing selfishly constructed overlay networks, in *In Proceedings of IEEE INFOCOM'04, Hong Kong*, March 2004.

[8] A. Soule, K. Salamatian, A. Nucci, and N. Taft, Traffic matrix tracking using kalman filters, in *SIGMETRICS Perform. Eval. Rev.*, New York, NY, USA, 2005.

[9] M. Youssef, C. Scoglio, and T. Easton, Optimal topology design for overlay networks, in *IFIP Networking 2007*, 2007.

[10] S. Bhandari, B. Choi, and E. Park, Hybrid topology for multicast support in constrained wdm networks, in *International Teletraffic Congress (ITC)*, 2007.

[11] H. Zhang, J. Kurose, and D. Towsley, Can an overlay compensate for a careless underlay?, in *Proceedings of IEEE INFOCOM'06*, Barcelona, Spain, April 2006.

[12] J. Fan and M. Ammar, Dynamic topology configuration in service overlay networks: A study of reconfiguration policies, in *Proceedings of IEEE INFOCOM'06*, 2006.

[13] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker, On selfish routing in internet-like environments, in *Proceedings of the ACM SIGCOMM, series = All ACM Conferences, pages = 151–162, month = august, year = 2003,*.

[14] B. McBride and C. Scoglio, Characterizing traffic demand aware overlay routing network topologies, in *Workshop on High Performance Switching and Routing*, 2007.

[15] Z. Li and P. Mohapatra, On investigating overlay service topologies, in *Computer Networks*, volume 51, pages 54–68, 2007.

[16] A. Fabrikant, A. Luthra, E. Maneva, C. Papadimitriou, and S. Shenker, On a network creation game, in *in Proceedings of ACM PODC*, 2003.

[17] M. Pioro and D. Medhi, Routing, flow, and capacity design in communication and computer networks: Chapter 4, in *Morgan Kaufmann*, 2004.

[18] S. Chen, F. Boesch, and J. McHugh, On covering the points of a graph with point disjoint paths, in *Graphs and Combinatorics (Proc. Capitol Conf. on Graph Theory and Combinatorics)*, 1974.

[19] K. Zhu, Us nation-wide ip backbone network topology.

[20] N. Spring, R. Mahajan, and D. Wetherall, Measuring isp topologies with rocketfuel, in *Proceedings of ACM SIGCOMM'02*, 2002.

[21] M. Faloutsos, P. Faloutsos, and C. Faloutsos, On power-law relationships of the internet topology, in *SIGCOMM*, pages 251–262, 1999.

[22] P. Mahadevan et al., Lessons from three views of the internet topology, in *Tech. rep., Cooperative Association for Internet Data Analysis (CAIDA)*, 2005.

[23] C. Xie, S. Guo, R. Rejaie, and Y. Pan, Examining graph properties of unstructured peer-to-peer overlay topology, in *Proceedings of the 10th IEEE Global Internet Symposium (GI'07)*, 2007.

[24] S. Dorogovtsev, Networks with given correlations, in *www.arxiv.org*.

[25] B. McBride, C. Scoglio, and S. Das, Distributed biobjective ant colony algorithm for low cost overlay network routing, in *Proceedings of ICAI 2006*, Las Vegas, USA, 2006.

[26] S. Banerjee, D. Tipper, M. Weiss, and A. Khalil, Traffic experiments on the vbns wide area atm network, in *IEEE Communications*, pages 126–133, 1997.

[27] K. Papagiannaki, N. Taft, and A. Lakhina, A distributed approach to measure ip traffic matrices, in *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, 2004.

[28] A. Soule et al., Traffic matrices: balancing measurements, inference and modeling, in *SIGMETRICS Perform. Eval. Rev.*, New York, NY, USA, 2005, ACM Press.

[29] A. Soule, A. Nucci, R. Cruz, E. Leonardi, and N. Taft, How to identify and estimate the largest traffic matrix elements in a dynamic environment, in *SIGMETRICS '04/Performance '04: Proceedings of the joint international conference on Measurement and modeling of computer systems*, New York, NY, USA, 2004, ACM Press.

[30] Abilene, Abilene internet 2: www.abilene.internet2.edu.

[31] L. Sahasrabuddhe and B. Mukherjee, Light-trees: Optical multicasting for improved performance in wavelength-routed networks, in *IEEE Communications Magazine*, 1999.

[32] E. Varma, S. Sankaranarayanan, G. Newsome, Z. Lin, and H. Epstein, Architecting the services optical network, in *IEEE Communications Magazine*, 2001.

[33] B. Mukherjee, Optical communication networks: Wdm, broadcast/multicast and wavelength-routing, in *Mc Graw Hill*, 1997.

[34] T. Stern and K. Bala, Multiwavelength optical networks, in *Addison Wesley*, 1999.

Appendix A

Problems formulations for WDM Optical Networks

A.1 Adding and Dropping physical and virtual links from the existing multicast sessions

The goal of this formulation is to reconfigure the existing multicast session topologies to satisfy the new multicast demands and members.

A.1.1 Data Input

- $padjacent_{m,n}$: The adjacency matrix of the physical topology. $padjacent_{m,n} = 1$ if there is a link between m and n nodes.

- $vadjacent_{m,n}$: The adjacency matrix of the virtual topology. $vadjacent_{m,n} = 1$ if there is a link between m and n nodes.

- $M_{i,m,n}$: The adjacency matrix of the physical topology for the session i.

- $Y_{i,m,n}$: The adjacency matrix of the virtual topology for the session i.

- $newsourse_i$: It contains the source nodes of the new sessions.

- $newsession_{i,m}$: It is a binary data containing the node members of the sessions. $newsession_{i,m} = 1$ if for session i the node m is a member.

- $w_{m,n}$: Link weight of the physical topology.

- α: Link weight of the virtual topology.

- cp: number of available channels on each physical link.

- cv: number of available channels on each virtual link.

A.1.2 Decision Variables

- $px_{i,m,n}$: It is a binary decision variable. $px_{i,m,n} = 1$ if a physical link between nodes m and n is included in the physical topology for the existing session i to satisfy the new members.

- $vx_{i,m,n}$: It is a binary decision variable. $vx_{i,m,n} = 1$ if a virtual link between nodes m and n is included in the virtual topology for the existing session i to satisfy the new members.

- $pz_{i,m,n}$: It is a binary decision variable. $pz_{i,m,n} = 1$ if a physical link between nodes m and n is dropped from the physical topology for the existing session i to satisfy the new members.

- $vz_{i,m,n}$: It is a binary decision variable. $vz_{i,m,n} = 1$ if a virtual link between nodes m and n is dropped from the virtual topology for the existing session i to satisfy the new members.

- $pw_{i,m,n}$: It is a binary decision variable. It represents the final physical topology of the new session i after adding and dropping links from the existing session i.

- $vw_{i,m,n}$: It is a binary decision variable. It represents the final virtual topology of the new session i after adding and dropping links from the existing session i.

- $f_{i,m,n}$: It represents the amount of flow over the physical link (m, n) for session i.

- $y_{i,m,n}$: It represents the amount of flow over the virtual link (m, n) for session i.

A.1.3 Objective

$$minimize \sum_i \sum_m \sum_n w_{m,n} pw_{i,m,n} + \alpha vw_{i,m,n} + px_{i,m,n} + vx_{i,m,n} + pz_{i,m,n} + vz_{i,m,n} \quad (A.1)$$

The objective is composed of two parts. The first objective is to minimize the overall cost of the physical and virtual link weights. The second is to minimize the total number of added and dropped physical and virtual links between the existing and the new sessions.

A.1.4 Constraints

$$pz_{i,m,n} \leq M_{i,m,n} \; \forall i \; m \; n \quad (A.2)$$

This equation means that if the physical link between nodes m and n is included in the physical topology for the existing session i i.e. $M_{i,m,n} = 1$, it may not be included (dropped) in the configuration for the new session i.e. $pz_{i,m,n} = 1$

$$vz_{i,m,n} \leq Y_{i,m,n} \; \forall i \; m \; n \quad (A.3)$$

It is the same as in A.2 but for virtual links.

$$px_{i,m,n} + pz_{i,m,n} \leq 1 \; \forall i \; m \; n \quad (A.4)$$

This equation avoids including and dropping the physical link between nodes m and n for session i simultenously.

$$vx_{i,m,n} + vz_{i,m,n} \leq 1 \; \forall i \; m \; n \quad (A.5)$$

It is the same as in A.4 but for virtual links.

$$px_{i,m,n} + M_{i,m,n} - pz_{i,m,n} = pw_{i,m,n} \; \forall i \; m \; n \quad (A.6)$$

The variable $pw_{i,m,n} = 1$ when either the physical link (m, n) presented in the topology for current session i ($M_{i,m,n} = 1$) and it is also presented in the new session topology, or the physical link

(m, n) is not presented in the existing topology ($M_{i,m,n}$=0) and this link becomes a part of the new configuration for the new session ($px_{i,m,n} = 1$). This variable can also equal to zero when a physical link presented in the existing session i ($M_{i,m,n} = 1$) and it is no longer a part of the current configuration for that session ($pz_{i,m,n} = 1$).

$$vx_{i,m,n} + Y_{i,m,n} - vz_{i,m,n} = vw_{i,m,n} \; \forall i \; m \; n \tag{A.7}$$

It is the same as in A.6 but for virtual links.

$$\sum_{n} y_{i,newsource_i,n} + f_{i,newsource_i,n} = \sum_{k} newsession_{i,k} \; \forall i \tag{A.8}$$

Source node of each session sends the traffic demand to all the destination using both the virtual and physical links attached with the source node.

$$\sum_{m} y_{i,m,n} - y_{i,n,m} + f_{i,m,n} - f_{i,n,m} = newsession_{i,n} \; \forall i \; n \neq newsource_i \tag{A.9}$$

It is the flow conservation flow equation.

$$\sum_{i} f_{i,m,n} \leq padjacent_{m,n} C \; \forall m \; n \tag{A.10}$$

If there is a physical link (m, n), flow can flowed over it for any session. C is a big number.

$$\sum_{i} y_{i,m,n} \leq vadjacent_{m,n} C \; \forall m \; n \tag{A.11}$$

It is the same as in A.10 but for virtual links.

$$f_{i,m,n} \leq pw_{i,m,n} \sum_{k} newsession_{i,k} \; \forall i \; m \; n \tag{A.12}$$

This equation means if $pw_{i,m,n} = 1$, it may be used to send the flow over it for session i.

$$\sum_{i} pw_{i,m,n} \leq cp \; \forall m \; n \tag{A.13}$$

This equation constrains number of available channels on each link.

$$y_{i,m,n} \leq vw_{i,m,n} \sum_{k} newsession_{i,k} \; \forall i \; m \; n \tag{A.14}$$

$$\sum_{i} vw_{i,m,n} \leq cv \; \forall m \; n \tag{A.15}$$

Equations A.14 and A.15 are similar to equations A.12 and A.13 but for virtual links respectively.

A.2 Simple Reconfiguration ILP formulation

This ILP formulation has the same function as in section A.1 but it is simpler. It has the same data input and subset of the decision variables and subset of the constraints.

A.2.1 Decision Variable

$pw_{i,m,n}$, $vw_{i,m,n}$, $f_{i,m,n}$ and $y_{i,m,n}$.

A.2.2 Objective

$$minimize \sum_{i}\sum_{m}\sum_{n} w_{m,n}pw_{i,m,n} + \alpha vw_{i,m,,n} + abs(M_{i,m,n} - pw_{i,m,n}) + abs(Y_{i,m,n} - vw_{i,m,n}) \tag{A.16}$$

This objective equation is similar to A.1 where the first part is minimizing the cost of constructing the session topologies while the second is minimizing the absolute difference between the existing session and the new session topologies (minimizing number of added and dropped physical and virtual link).

A.2.3 Constraints

Equations from A.8 to A.15 represent the constraint equations for this formulation.

A.3 Establishing new multicast sessions beside the existing sessions

In this formulation, given the existing session and we would like to include new sessions based on the available channels on the physical and virtual links.

A.3.1 Data Input

The same data input except for $newsession_{i,m}$ which is a binary data representing the members m in the session i and $newsourse_i$ which represents the source node of the $newsession_{i,m}$.

A.3.2 Decsion Variables

- $pw_{i,m,n}$: Binary variable representing the chosen physical link (m, n) in the new session i.
- $vw_{i,m,n}$: Binary variable representing the chosen virtual link (m, n) in the new session i.
- $f_{i,m,n}$
- $y_{i,m,n}$
- $pcapacity_{m,n}$: Number of available channels on the physical link (m, n).
- $vcapacity_{m,n}$: Number of available channels on the virtual link (m, n).

A.3.3 objective

$$minimize \sum_{i} \sum_{m} \sum_{n} w_{m,n} pw_{i,m,n} + \alpha vw_{i,m,,n} \tag{A.17}$$

The objective is to minimize the cost function of constructing the topologies for the new sessions.

A.3.4 Constraints

$$(cp - \sum_{i} M_{i,m,n}) padjacent_{m,n} = pcapacity_{m,n} padjacent_{m,n} \; \forall \; m \; n \tag{A.18}$$

This equations calculates the available channels on the physical links given the existing session physical topologies.

$$(cv - \sum_i Y_{i,m,n})vadjacent_{m,n} = vcapacity_{m,n}vadjacent_{m,n} \; \forall \; m \; n \tag{A.19}$$

This equation has the same function as A.18 but for virtual links.

Equations A.8, A.9, A.10, A.11, A.12 and A.14 are the same.

$$\sum_i pw_{i,m,n} \leq pcapacity_{m,n} \; \forall m \; n \tag{A.20}$$

This equation constrains number of the new sessions i over the physical link (m, n) according to the available channels $pcapacity_{m,n}$.

$$\sum_i vw_{i,m,n} \leq vcapacity_{m,n} \; \forall m \; n \tag{A.21}$$

This equation has the same function as A.20 but for virtual links.