Evaluation of heuristics towards a solution to the SDM routing problem

Bachelor's Project Computing Science

July 2024

Author: Nevis King
Student Number: S4400771
First supervisor: Reza Zare Hassanpour
Second supervisor: Fadi Mohsen

Abstract

Serving the ever increasing amount of internet traffic that the information age necessitates, demands improvements in network transfer speed. One solution to this problem is found in Spacial Division Multiplexing, a technique for allowing multiple data streams to be sent over the same optical fiber. However this technology is not a drop in replacement for current infrastructure and its advantages come with limitations in how these data streams can be routed. Fully utilizing this technology requires finding efficient solutions for the routing problem in the context of spacial continuity. Towards finding such a solution, in this project I have examined heuristics to be used in Spacial Division Multiplexed routing, and analyzed their effects across different network topologies and requirements.

Contents

List of Figures

List of Tables

1 Introduction

The aim of this paper is to lay a groundwork for creating solutions to the RSCSA problem. SDM routing breaks some of the fundamental assumptions made in typical network routing. As such, in order to use these assumptions, they must first be tested to be correct. We will take some of these assumptions and simulate them across a diverse set of situations to see if they still hold in the SDM domain. The main challenge of the RSCSA problem is that routing algorithms used for typical networks cannot be used. We need to re-evaluate network algorithms from first principals.

Within the scope of this paper are problems related to the selection of light paths for messages as well as selecting which spacial layer these light paths should be on. We concern ourselves with what heuristics prove beneficial in reducing the total number of time steps required to serve all messages on a network. This leaves some other aspects of network quality out of our scope. Notably we do not look into message turnaround time nor time spent in transfer. It is entirely possible for the first part of a message to be sent hundreds of time steps before its last part. In order to discover what heuristics are beneficial we have created a simulation which replicates the physical layer of a SDM network. We have implemented various heuristics in this simulation which can be enabled or disabled, and have recorded the effects these heuristics have on total solution time.

The heuristics we analyze in this paper fall into two categories. The first category is that of route selection heuristics. Specifically we test hierarchical route selection, random route selection, and shortest route selection. Shortest route selection is the typical standard in usual network routing. The others are introduced as points of comparison so as to evaluate if shortest path routing is still effective in SDM. The implementation of these heuristics will be expanded on later. The second category is the own contribution of this paper, we refer to it as temporal batching. Rather than sending messages as soon as possible, we group messages that arrive by source destination pair and send them only when they can fill a lane. This heuristic can be used in combination with any of the routing selection heuristics so we analyze them in all permutations. The implementations of these heuristics as well as the simulation are expanded on in the Models and Algorithms section.

1.1 Aim of the Paper

This paper endeavours to provide the groundwork on which future research attempting to solve the dynamic RSCSA can build on. It can be categorized best under the field of Network Engineering, specifically it is concerned with the physical layer of optical networks. The research question that the RSCSA problem leads this paper to is ”What heuristics are beneficial in the Dynamic RSCSA”. Our goal is to propose some such heuristics and then evaluate their utility on SDM networks.

1.2 Proposal

There are many preexisting papers which present approaches and even solutions to the static RSCSA problem. Their solutions however cannot be simply extended to the dynamic RSCSA problem. Little research has been done into the dynamic RSCSA problem due to the exponential increase in complexity it has over the static problem. We belive that by finding heuristics which lead to better outcomes in SDM routing, we can decrease the total search space that future research in answering the dynamic RSCSA will need to consider. By establishing some first principles we can validate that some assumptions made about routing which come from non-SDM domains still do or do not hold true.

1.3 Motivation

The discovery of such heuristics would be highly impactful on the future of SDM networking. The high capacity data transfer which SDM enables is constrained by the limitation which the technology introduces. Finding heuristics which can capture more of the advantages that SDM networking can confer would significantly contribute to making SDM networking more cost efficient, and thereby increase the utility of the technology.

1.4 Paper Structure

Section 2 summarizes relevant background information necessary for understanding Wavelength Division and Spacial Multiplexing. Section 3 goes on to cover some relevant related research which has been conducted which this paper draws heavily from. Section 4 goes over the implementation of generative data which will be used for testing our heuristics. Section 5 details the implementation of the SDM simulation and the choices made and their rationale. Section 6 discusses the results of the simulations and our findings. Section 7 concludes the paper and offers a summary of the findings. Section 8 has this papers acknowledgements. Section 9 has a detailed tables of the simulation results. Refrences to other works can be found at the end of the document.

2 Background

In 2022 the global bandwidth usage was 1,200 Terabits per second [2]. As demand for point to point data transfer has grown, so has the infrastructure which supports it. Breakthroughs in optical fiber technology are what made the transfer speeds we enjoy now possible. In 2006 researchers at the Nippon Telegraph and Telephone cooperation achieved rates of data transfer of over 100 Gigabytes per second over a single optical fiber [1]. With the advent of Wave Division Multiplexing in 2013, the NEC Corp. of America were able to send data at rates of 1.05 Petabytes per second [4]. In 2023 scientists at the National Institute of Information and Communication Technology of Japan achieved an astounding transfer speed of 22.9 Petabytes per second using Spacial Division multiplexing [3]. This incredible new technology has its own restrictions which will need to be addressed to best utilize it.

The fundamental problem that Space Division Multiplexing (SDM) introduces is that it has additional restrictions on its routing that previous technologies didn't have. This problem of routing in the SDM domain is referred to as the Routing, Spatial Channel, and Spectrum Assignment (RSCSA) problem [7]. Calculating the most efficient way to route messages in SDM even in the static context has been shown to be a NP-hard problem [7]. This means that for real world applications some kind of heuristic must be employed. Ideally this heuristic would approach the efficiency of an optimal solution without nearly as much computational demand. In this paper I will propose heuristics to be employed in the dynamic case of SDM routing and will analyze their utility.

2.1 Wavelength division multiplexing

Wave division multiplexing (WDM) is a technique in optical networking which allows multiple streams of information to be transferred across the same optical fiber without interference [6]. WDM works by transmitting each stream as a wave of light in a different frequency [6]. Optical Cross Connects (OXC) form the backbone of networking in the WDM domain. OXC allow for a given wavelength of light on any of its input ports to be routed independently of any other wavelengths to one of its output ports [6]. This allows for the construction of arbitrary mesh topologies taking advantage of the increased bandwidth allowed for by WDM [6].

2.2 Routing in the Wavelength Domain

Lightpaths are the series of hops across OXC on a given optical mesh network that a message takes from its source to its destination [6]. Much like in non-optical networking, an optical network must assign paths to messages to allow them to reach their destination. Typically the routing problem involves finding a path through a mesh network. In the wavelength domain the problem is made more complex because of the two constraints identified by G.N. Rouskas and H.H. Perros. The first of these constraints is the Wavelength continuity constraint, which states that a lightpath must be assigned a certain wavelength which is unchanged from source to destination. The second is the Distinct wavelength constraint, which states that every lightpath which travels along the same fiber must be allocated a distinct wavelength. This effectively duplicates the topology of the mesh into as many separate sub-meshes as the network supports frequencies [6]. This means solving the routing of a lightpath may be equivalent to solving as many routing problems as the network has frequencies [6].

A potential solution to this computational bottleneck can be found in Wavelength Converters, a single input single output device which can convert the wavelength of an optical signal [6]. Combining full-spectrum converters with OXC has the advantage of relaxing the Wavelength continuity constraint, transposing the wavelength routing problem back into the typical routing problem [6]. However this is only so simple when considering the case where every OXC is equipped with a full spectrum converter, if only some are equipped or if they are equipped with limited wavelength converters then the routing algorithm will need to be adjusted [6].

2.3 SPACE DIVISION MULTIPLEXING

Space division multiplexing (SDM) is another method of increasing the bandwidth of an optical fiber [5]. In SDM optical fibers are manufactured in such a way that they have multiple cores through which optical signals are guided [5]. This allows multiple signals on the same wavelength to be sent through the same fiber on different cores [5]. SDM comes in two theoretical forms, the cross-talk and the no cross-talk forms. In the no cross-talk form each core is perfectly isolated meaning there are no additional considerations to using SDM in a given optical network. In the more realistic cross-talk form the cores have some amount of interference with each other, meaning the signal will have to be analyzed to transform it back into its constituent signals [5]. This poses a problem when it comes to optical network routing in that the group of cores must therefore be routed to the same analyzer as otherwise the original signals will not be recoverable [5]. Using cross-talking SDM therefore poses additional complexity on top of the complexity inherent to WDM routing. In this paper we assume we are utilizing SDM fibers without cross-talk, as our concern is with the routing of SDM signals and not the mechanical design of optical fibers. This leaves us with the assumption that signals on different spacial layers can be routed independently of each other.

2.4 SDM/WDM ROUTING

A key issue with pure SDM is that the entire spacial lane needs to be routed from a single source to a single destination [7]. This problem can be alleviated with the application of Wavelength Cross Connects (WXC), a variant on the OXC which allows for both wavelength based routing as well as wavelength switching. WXC allow for adding or dropping signals to Spacial lanes, meaning that every signal on the lane no longer needs to have the same source destination pair [7]. This means that when considering routing, in the case that every node in a network has a WXC for a given spacial layer, the common source destination requirement can be relaxed. However building a spacial lane with WXC is inherently more expensive than building it without one, so it becomes necessary to develop a routing algorithm which can minimize the number of WXC layers a network requires while still performing adequately [7].

3 Related Work

The primary inspiration for this paper is *Hierarchical Routing and Resource Assignment in Spatial Channel Networks (SCNs): Oriented Toward the Massive SDM Era* written by Mingcong Yang, Maiko Shigeno, and Yongbing Zhang [7]. Not only does it coin the RSCSA problem this paper deals with, it also provides a solution to the static RSCSA problem. They prove the NP hardness of the RSCSA problem in their thirds section. Their solution for the static RSCSA problem uses an arbitrary path selection which inspired this paper's testing of different path selection heuristics. Their use of heuristics which attempt to fill spacial lanes as much as possible inspired this paper's temporal batching heuristic. Specifically they solve the static RSCSA for a single time step, leading directly to the question "if these choices are optimal for a single time step, what choices would be optimal for multiple time steps".

In their paper the authors define a number of connection types, messages which can occupy a full spacial lane (Type 1), messages which can share a lane with other messages (Type 2), messages on a wavelength switching band (Type 3), and messages which occupy multiple full bands (Type 4) [7]. Their paper defines a set of algorithms which takes in message requests one at a time [7]. Our paper treats all message requests in the same way, but our batching heuristic can be seen as a way of turning their type 2 messages as type 1 messages. The type 4 messages are not covered by our paper, they are simply split into smaller messages and routed as normal. Their type 3 messages are identical to the messages routed on wavelength switching layers in this paper. This paper can be seen as a continuation of their paper into the dynamic domain.

4 Materials and Methods

4.1 Data

The data for this paper was generated with a custom script written for this project in Rust. This script takes in parameters for lane width, number of layers, number of wavelength switching capable layers, number of nodes, an identifier, an average amount of data units sent per time step, and a network shape. The output of this script takes the following form.

```
1  // 3_3_3_5_ring_1_20.txt
2  layers: 3, switching: 3, width: 3
3  link 0 to 1
4  link 1 to 2
5  ...
6  link 4 to 0
7  0 to 1, size 4, time 0
8  0 to 3, size 9, time 0
9  0 to 2, size 7, time 0
10 2 to 4, size 19, time 1
11 0 to 1, size 1, time 1
12 ...
13 4 to 3, size 1, time 98
14 2 to 3, size 1, time 98
15 2 to 1, size 10, time 99
16 4 to 1, size 6, time 99
17 1 to 4, size 4, time 99
```

Here line 1 is the name of the file, it has all relevant parameters stored in its name for easy identification as well as an id tag to separate files with the same parameters but different randomized contents. Line 2 defines the number of layers, as well as the width of a lane, that being how many data units one lane can transfer in one time step. It also specifies the first layer (by 0 index) which is capable of wavelength switching. All layers which have an equal or greater index are assumed to be wavelength switching. In this example the network has no wavelength switching layers. Lines 3-6 show how links are specified, with a pair of node Ids. Duplicates can exist in this list and are handled gracefully by the simulation (only the first instance of a pair is registered), this also applies to duplicates of a reversed order. That is to say data file with the following lines:

```
link 0 to 1
link 1 to 0
link 1 to 0
```

Would define a network with exactly two nodes and only a single connected lane between them. Lines 7-17 make up the bulk of the data file and are the representations of messages which are requesting to be sent on the network. A message is represented with three components, a source destination pair of node Ids, a size in data units, and a time of arrival. Message requests are generated in the time-bounds from 0-99, although the simulation could run with messages from any time bounds. The selection of this boundary is arbitrary as the important thing to be able to compare the simulation across different datasets is the time the data can be transferred in compared to the time it took to send. No matter what the upper bound of the message arrival time is, a simulation which has sufficient capacity could finish sending all the messages in precisely that many time steps. So 100 total time steps is selected so as to give a common ground to compare results across. The messages are listed in order of arrival although that is also not necessary for the simulation, it takes into account the written arrival time. Although the ordering within a single time step does affect in which order the simulation will try to resolve the message requests. This means it is not sufficient to test the simulation on a single data file with a given set of parameters, instead multiples with the same parameters need to be generated and their results averaged together. This is why the id parameter exists.

4.2 Data generation

Data file are generated in three main parts, the network specifications, the topology, and the message requests. First the network specifications can be read directly from the input parameters, these are the first three numbers of the data file's name. The topology is generated based on the fourth and fifth arguments to the generating script. These are the number of nodes and the topology shape respectively. The topology shapes are defined as follows. A ring topology consists of a series of nodes which are connected to the node one index ahead of them, and one index behind them. For the first and last node this wraps around meaning they have a connection. For a ring topology of size three this would look like:

```
link 0 to 1
link 1 to 2
link 2 to 0
```

A scatter topology defines a ring topology with additional connections. Each node is connected to the same nodes it would be in a ring topology but with an additional 0-2 connections to random other nodes. The minimum size for a scatter topology to differentiate itself from a ring topology is 4. Scatter topologies offer more interesting routing problems than ring topologies. In a ring topology there are a maximum of two paths to get from any given node to any other, clockwise or anti-clockwise around the ring. In a scatter topology no such guarantees about the number of paths can be made without calculating each of them. The additional links and routes provided by a scatter topology increase the search space for routes dramatically. This comes with the downside that since our simulation considers all possible routes, there is an upper bound of how large our scatter topologies can be before we cannot realistically run our simulations on consumer hardware.

The sixth argument is an Id specifier which is carried over the the name of the output file. Finally the seventh argument specifies the number of data units the message should generate per time step. When generating message requests the script will choose a random number from 1 to the remaining data it needs to send. A message will be generated for a randomly selected source destination pair with the aforementioned size. This will repeat until the script has added the specified amount of data for the time step. Then it repeats this process for the other 99 time steps.

4.3 Data selection

To test our heuristics it is imperative to select diverse network structures as well as message densities, as some heuristics may perform very well on a ring topology but perform horribly on a scatter topology or vice versa. Additionally quantity of nodes in a network must be considered, although with the limitation that generating results for large networks scales poorly. To account for fluctuations in efficiency due to the order in which messages arrive we will also opt to generate multiple instances of each data source.

Since we would like to be able to compare the efficiencies of heuristics on different data sources while also evaluating their combinations it will become necessary to evaluate each heuristic in each permutation with the others as well as with each permutation of the network. To limit the scope of this undertaking we will define for each network parameter a low, medium, and high value. For each network heuristic there is a boolean on or off. For network size we will define low as 3 nodes, medium as 6 nodes, and high as 9 nodes. High in this sense refers to the upper bounds of what we are able to simulate in a reasonable amount of time, not what would be a high network size in the real world. For the inputs of lane width, number of layers, and data per time-step they are all connected, a network with small lanes but little traffic is comparable to a network with large lanes and high traffic. So combining there parameters we define the combined ”network capacity” as low when having 3 spacial layers, 3 data unit wide lanes, and 20 data units per time step per node. This is roughly twice as much data per time step as the network could transfer under ideal conditions. For a medium capacity we use 3 spacial layers, 5 data unit wide lanes, and 20 data units per time step per node. This gives the network a bit more data per time step than it could transfer under ideal conditions. Finally the high network capacity is defined as 4 spacial layers, 5 data unit wide lanes, and 20 data units per time step per node. This gives the network enough capacity to transfer all the data each time step under ideal conditions. We only test up to the rate that the network could process under ideal conditions because testing values for which the network completes the transfer in the minimum possible time (100 time steps) makes comparing their efficiency impossible. Finally we declare a small, medium, and high amount of wavelength switching capable layers as 0, 1, and 2. We choose these numbers because as previously specified our number of layers are always either 3 or 4. This means our high end represents a network where a majority of spacial lanes are wavelength switching, our low end shows results of not allowing wavelength switching, and our medium point allows exactly one wavelength switching lane.

4.4 Technology Stack

The programming language used to implement the SDM simulation was Rust. Testing was implemented with Rust’s native testing functionality. Useful data structures such as Queues were imported from Rust’s standard library. Likewise Rust’s standard library was uses for file IO. For generating random data the Rand package was used. All other functionality including network simulation was written for this project.

5 Models and Algorithms

Given that this paper is concerned with the evaluation of certain heuristics on the performance of SDM networks, clearly we require either access to such a network or a reasonable simulation of one. We took the latter approach. The simulation follows the rules of real SDM networks although it uses some abstractions and assumptions to narrow the scope of the project. This simulation is principally concerned with simulating the routing of SDM networks, not the administration of the network itself. As such the compute time for a route, which is very relevant in real world application, is not investigated. This paper is also not concerned with flow of control commands between routers, we simulate specifically the flow of message data through the network on a physical level, higher level network organization is outside of the scope of this paper.

5.1 Lanes

The central structure of this simulation is called a lane. A lane represents a single spacial lane connecting two nodes. One property of a lane is its width, a lane's width which denotes how many Data-Time Units (DTU) a lane has capacity for. A DTU represents the amount of data which the network can transmit over a wavelength slice in one time step. The length of a time step, the size of a wavelength slice, and the corresponding DTU size is all left as undefined as they would be specific to the physical construction of a given network. The lane has a fixed length vector which represents the lane across wavelength space. The length of this vector is precisely the width of of the lane. This vector is updated throughout the simulation to keep track of which wavelength slices of the lane are occupied or free. A free wavelength slice is represented by a 0 at that index of the vector. Any other number represents a message with that ID being transferred on that lane. Finally a lane may have a source and destination pair. If it does, this pair represents not the two nodes which the lane connects, but rather the routing pair which the lane is locked to, this will be elaborated on in the subsection *Routing*.

5.2 Layers

A layer represents a slice of the network along the spacial axis. At first it may seems strange to subdivide our network this way as it means cores within the same optical fiber are being considered as being on separate layers, but when considering the properties of SDM routing the reasons for doing this becomes clear. Since messages cannot switch spacial channels while they are being transferred, and cannot be added or dropped to/from an occupied lane unless it is wavelength switching capable, treating the different spacial channels as wholly disconnected layers is a useful abstraction. A layer is composed of a single object, a hashmap which binds a source destination pair of nodes to a lane. Routing operations are always applied to a single layers, and all layers are treated as independent of the others. The only additional property that a layer can have is whether or not it is a wavelength switching layer. For a layer to be a wavelength switching layer every spacial lane on that layer must be wavelength switching capable. This allows some of the routing restrictions imposed by SDM routing to be relaxed. In the case of simulating a physical network where some spacial lanes on a given layer are wavelength switching capable but not all, this must be considered as a non wavelength switching capable layer for the purposes of this paper. This is because the way wavelength switching layers are used in this paper require absolutely every lane to be wavelength switching capable. This also means that in the case of modeling a real network it may be best to underestimate the number of wavelength switching capable layers it has as a single link in that layer malfunctioning would change the status of the whole layer.

5.3 Messages

Messages in the simulation are modelled with a few components, first a unique identifier, a source destination pair, and a size. The size of a message is in DTU, this means that it is easy to see if a given lane has the capacity to carry a message. If a message is 3 DTU and a lane has 4 wavelength slots open, the lane has enough capacity. Finally a message has an arrival time, this is used for calculating the turnaround time of a message as well as for specifying at what time step a message should start being considered.

5.4 Network

The network itself is comprised of the previously explained components. First it holds some metadata related to the width of lanes within the network, the number of layers, and a counter for generating new message identifiers. It stores a vector of layers, representing the spacial layers of the network. It stores all messages which need to be handled in a message queue. This is a simple queue object where each component is a message and they are in ascending order of arrival time. It also has a corresponding batch queue with the same structure, more on that in the *Heuristics* section. A vector of Source Destination pairs is stored representing all links in the network. Notably both directions of the pair are stored, this allows for faster checking in the layer's hashmap. Finally a layer switching index is recorded. This index denotes which index of the layer vector should be the first one to be considered as wavelength switching capable. It is assumed that all layers stored on higher indices are also wavelength switching capable.

5.5 Routing

In this paper we are concerned with the routing efficiency in a SDM network, not the computational efficiency of a routing algorithm. For this reason we want to be able to compare all possible routes that a message could take. For the sake of clarity moving forward a route will refer to the series of hops across nodes that a message could take to its destination. Meanwhile a path will refer to the series of hops a message has taken, bundled with the wavelength slices it occupies and the layer it has been sent on.

To find which routes are available to the message, the simulation uses an exhaustive recursive search. Passing forward a vector of all nodes which have been visited, the function calls itself on all connected nodes which have not yet been visited. If the currently visited node is the destination node, the function returns the route that has been taken to reach it, along with a vector with the utilization information of every lane along the way. If the visited node is not the destination node, the concatenation of the results of the recursive calls are returned instead. These results are then crawled to find all valid paths, that is paths which have sufficient capacity for the message as well as not being reserved by any other source-destination pairs. Of these valid paths one is selected by whichever path selection heuristic is active.

5.6 Heuristics

Our heuristics fall into two categories, route selection heuristics and the temporal batching heuristic. The route selection heuristics are a diverse selection of route selection strategies which can be found in other types of network routing, while the temporal batching heuristic is an own contribution of this paper. Temporal batching refers to batching messages of common source destination pairs into *batched* messages of a size equal to the width of a lane. These batched messages are treated the same as any other message for the purpose of the simulation. Their purpose is to strive to fully occupy a spacial lane if it is to be used. If the batching heuristic is used, batched messages will be constructed from available messages if possible and these messages will be given priority when assigning routes over non batched messages. The set of route selection heuristics are all mutually exclusive, only one can be enabled at any one time, and at least one must be enabled. The first route selection heuristic is hierarchical selection. This heuristic assigns each lane a unique priority in the order they are specified in the input file. When multiple valid routes are available, the route with the highest hierarchical order will be selected (the next lane being checked in case there are multiple which share a common lane and so on). Hierarchical routing has an advantage in that lanes which it is assumed less messages will need to use can be assigned higher priority to reserve more space on more important lanes. Shortest path selection is the next heuristic and is one which is relevant in other network routing domains. It is included as a heuristic here as the limitations of SDM routing mean that shortest path selection can actually result in sub optimal routing assignments. It is presumed that shortest path routing will have the best results of the routing assignment heuristics, but it is still tested in this paper for completeness. The final routing selection heuristic is the random selection heuristic. It is included as a counterpart to the hierarchical and shortest path heuristics. It should balance loads more evenly through the network than either the shortest path (which will prioritize central lanes), or the hierarchical path (which will prioritize high priority lanes). If distributing paths more evenly is beneficial is an interesting factor to consider in SDM routing which would rarely ever be useful in other domains.

6 Results and Discussion

The individual results of our simulation can be found in the section *Simulation Results*. Here we will discuss specifically the results from Table 2 as they are the most intuitive to discuss. As our baseline we can see that Hierarchical path selection of course always has an efficiency of 1. All other heuristics are evaluated according to their relative efficiency compared to this heuristic. The relative efficiency of a heuristic on a given network is calculated as (1 + (Hierarchical time steps - Heuristic time steps) / Hierarchical time steps) and then rounded to two decimal places. To start with a sanity check we can see that for the Random path selection heuristic there is a scattering of efficiencies which go both above and below our baseline of 1. This is in line with what we expect, a random selection should be expected to sometimes outperform and sometime under perform. An interesting thing to note is that random selection seems to perform particularly badly on scatter networks, sometimes dropping to as low as 0.85 efficiency. However it consistently outperforms Hierarchical route selection on ring networks.

Moving on to our shortest path selection, it consistently outperforms every other path selection heuristic. This as well is to be expected, selecting the shortest path encodes the occupation of the least number of spacial lanes in the network for every given message. However comparing it to the hierarchical path selection is not entirely fair as our hierarchical implementation generates a random hierarchy. Under ideal circumstances hierarchical value assignment would happen manually or with some awareness of network traffic patters. In our simulation since message patterns are entirely random, hierarchical route selection does not have its opportunity to shine. This would be a good selection for future research, perhaps a dynamically adjusting hierarchical value assignment could be constructed for not random traffic patterns. The main concern with shortest path assignment being only a heuristic and not a probably best solution in this space seems to be assuaged, it still clearly outperforms other route selection heuristics, especially on scatter networks.

Finally we analyze the gains made by our own contribution heuristic, temporal batching. We can see that in every case our batching heuristic makes gains over the unbatched route selection heuristics. Over our baseline heuristic of hierarchical path selection our batching heuristic gives on average a 1.04 gain of efficiency. If we compare our random path selection with and without the batching we find a movement from 0.98 to 1.04. This implies that our batching heuristic makes some significant amount of gains even in circumstances where the route selection is sub-optimal. However the question remains if our heuristic is finding any amount of novel gains or if it is just capturing efficiency we could have found by using shortest path routing in the first place. When we look at the average shortest path routing efficiency we find it is 1.15, compared to our average shortest path batching efficiency of 1.19. This proves that our batching heuristic makes gains over even that efficiency captured by shortest path selection. Our results suggest that our heuristic can provide gains of roughly four percent across route selection heuristics and network types.

7 Conclusion

Returning to our research question: "What heuristics are beneficial in the Dynamic RSCSA", after thoroughly analyzing our data we conclude that we have identified at least two heuristics which fit this description. We have shown that the assumption that shortest path routing would still be efficient in SDM does in fact hold, despite the additional complications of SDM routing. We have also created our temporal batching heuristic and shown that it provides gains of about 4 percent even when combined with shortest path routing. We believe that by establishing these two core heuristics we have provided a valuable starting point for further research into the dynamic RSCSA problem.

The search space of solutions to the RSCSA problem is massive due to the complexity of the problem. Even just the work done in this paper could be extended in many different directions. For example adjusting the simulation to use non-random traffic patterns could provide very valuable insight into the effects of these heuristics on more realistic networks. Expanding the network size would also be interesting, our simulation topped out a 9 nodes with at most 3 connections, by using the assumption that shortest path routing will be used the efficiency of the simulation could be significantly improved. This would allow for gathering statistics on significantly larger network models. There are certainly heuristics which have not been considered in this paper which could contribute to more gains in efficiency.

8 Acknowledgments

I would like to thank Reza Zare Hassanpour, the supervisor of this project, for providing invaluable support and guidance on this topic. I would particularly like to thank him for giving me the freedom to follow the aspects of this topic which I found the most interesting.

I would also like to thank Mingcong Yang, Maio Shigeno, and Yongbing Zhang for their work on the paper which inspired this one [7].

9 Simulation Results

Network	Hierarchical	Shortest path	Hierarchical batching	Random	Shortest batching	Random batching
small_network-no_switch-low_capacity	328	302	321	321	292	315
small_network-no_switch-medium_capacity	196	172	187	189	166	184
small_network-no_switch-high_capacity	148	131	141	142	125	138
small_network-1_switch-low_capacity	327	303	319	321	291	316
small_network-1_switch-medium_capacity	195	175	188	189	164	183
small_network-1_switch-high_capacity	147	129	141	143	125	137
small_network-2_switch-low_capacity	326	301	319	320	286	313
small_network-2_switch-medium_capacity	195	176	188	188	166	184
small_network-2_switch-high_capacity	145	128	141	141	123	138
medium_network-no_switch-low_capacity_ring	644	605	623	624	571	605
medium_network-no_switch-medium_capacity_ring	379	333	352	357	315	343
medium_network-no_switch-high_capacity_ring	288	249	268	272	229	255
medium_network-no_switch-low_capacity_scatter	245	175	242	283	171	251
medium_network-no_switch-medium_capacity_scatter	172	142	168	201	138	183
medium_network-no_switch-high_capacity_scatter	126	103	118	135	101	121
medium_network-1_switch-low_capacity_ring	648	597	616	622	558	591
medium_network-1_switch-medium_capacity_ring	378	335	353	357	308	339
medium_network-1_switch-high_capacity_ring	287	257	270	274	241	261
medium_network-1_switch-low_capacity_scatter	290	249	288	315	245	294
medium_network-1_switch-medium_capacity_scatter	151	112	147	164	108	149
medium_network-1_switch-high_capacity_scatter	140	122	138	159	118	149
medium_network-2_switch-low_capacity_ring	638	589	616	615	558	592
medium_network-2_switch-medium_capacity_ring	380	349	361	366	323	348
medium_network-2_switch-high_capacity_ring	286	248	269	268	229	250
medium_network-2_switch-low_capacity_scatter	283	221	284	301	215	293
medium_network-2_switch-medium_capacity_scatter	151	112	147	164	108	149
medium_network-2_switch-high_capacity_scatter	127	103	124	134	101	126
large_network-no_switch-low_capacity_ring	968	875	924	918	833	887
large_network-no_switch-medium_capacity_ring	572	502	536	535	463	515
large_network-no_switch-high_capacity_ring	434	375	404	408	354	387
large_network-no_switch-low_capacity_scatter	395	322	394	470	319	425
large_network-no_switch-medium_capacity_scatter	212	149	209	259	143	226
large_network-no_switch-high_capacity_scatter	165	119	160	199	114	171
large_network-1_switch-low_capacity_ring	964	850	919	912	800	870
large_network-1_switch-medium_capacity_ring	567	506	533	535	467	510
large_network-1_switch-high_capacity_ring	430	364	393	404	333	368
large_network-1_switch-low_capacity_scatter	315	210	308	366	207	325
large_network-1_switch-medium_capacity_scatter	196	132	192	226	126	200
large_network-1_switch-high_capacity_scatter	147	107	143	166	104	145
large_network-2_switch-low_capacity_ring	962	871	916	921	817	879

Table 1: Network heuristic simulation results in time steps

Network	Hierarchical	Shortest path	Hierarchical batching	Random	Shortest batching	Random batching
small_network-no_switch-low_capacity	1	1.08	1.02	1.02	1.11	1.04
small_network-no_switch-medium_capacity	1	1.12	1.05	1.04	1.15	1.06
small_network-no_switch-high_capacity	1	1.11	1.05	1.04	1.16	1.07
small_network-1_switch-low_capacity	1	1.07	1.02	1.02	1.11	1.03
small_network-1_switch-medium_capacity	1	1.1	1.04	1.03	1.16	1.06
small_network-1_switch-high_capacity	1	1.12	1.04	1.03	1.15	1.07
small_network-2_switch-low_capacity	1	1.08	1.02	1.02	1.12	1.04
small_network-2_switch-medium_capacity	1	1.1	1.04	1.04	1.15	1.06
small_network-2_switch-high_capacity	1	1.12	1.03	1.03	1.15	1.05
medium_network-no_switch-low_capacity_ring	1	1.06	1.03	1.03	1.11	1.06
medium_network-no_switch-medium_capacity_ring	1	1.12	1.07	1.06	1.17	1.09
medium_network-no_switch-high_capacity_ring	1	1.14	1.07	1.06	1.2	1.11
medium_network-no_switch-low_capacity_scatter	1	1.29	1.01	0.84	1.3	0.98
medium_network-no_switch-medium_capacity_scatter	1	1.17	1.02	0.83	1.2	0.94
medium_network-no_switch-high_capacity_scatter	1	1.18	1.06	0.93	1.2	1.04
medium_network-1_switch-low_capacity_ring	1	1.08	1.05	1.04	1.14	1.09
medium_network-1_switch-medium_capacity_ring	1	1.11	1.07	1.06	1.19	1.1
medium_network-1_switch-high_capacity_ring	1	1.1	1.06	1.05	1.16	1.09
medium_network-1_switch-low_capacity_scatter	1	1.14	1.01	0.91	1.16	0.99
medium_network-1_switch-medium_capacity_scatter	1	1.26	1.03	0.91	1.28	1.01
medium_network-1_switch-high_capacity_scatter	1	1.13	1.01	0.86	1.16	0.94
medium_network-2_switch-low_capacity_ring	1	1.08	1.03	1.04	1.13	1.07
medium_network-2_switch-medium_capacity_ring	1	1.08	1.05	1.04	1.15	1.08
medium_network-2_switch-high_capacity_ring	1	1.13	1.06	1.06	1.2	1.13
medium_network-2_switch-low_capacity_scatter	1	1.22	1	0.94	1.24	0.96
medium_network-2_switch-medium_capacity_scatter	1	1.26	1.03	0.91	1.28	1.01
medium_network-2_switch-high_capacity_scatter	1	1.19	1.02	0.94	1.2	1.01
large_network-no_switch-low_capacity_ring	1	1.1	1.05	1.05	1.14	1.08
large_network-no_switch-medium_capacity_ring	1	1.12	1.06	1.06	1.19	1.1
large_network-no_switch-high_capacity_ring	1	1.14	1.07	1.06	1.18	1.11
large_network-no_switch-low_capacity_scatter	1	1.18	1	0.81	1.19	0.92
large_network-no_switch-medium_capacity_scatter	1	1.3	1.01	0.78	1.33	0.93
large_network-no_switch-high_capacity_scatter	1	1.28	1.03	0.79	1.31	0.96
large_network-1_switch-low_capacity_ring	1	1.12	1.05	1.05	1.17	1.1
large_network-1_switch-medium_capacity_ring	1	1.11	1.06	1.06	1.18	1.1
large_network-1_switch-high_capacity_ring	1	1.15	1.09	1.06	1.23	1.14
large_network-1_switch-low_capacity_scatter	1	1.33	1.02	0.84	1.34	0.97
large_network-1_switch-medium_capacity_scatter	1	1.33	1.02	0.85	1.36	0.98
large_network-1_switch-high_capacity_scatter	1	1.27	1.03	0.87	1.29	1.01
large_network-2_switch-low_capacity_ring	1	1.09	1.05	1.04	1.15	1.09

Table 2: Network heuristic simulation results in speed relative to Hierarchical routing

REFERENCES

[1] `https://web.archive.org/web/20170921122920/http://www.ntt.co.jp/news/news06e/0609/060929a.html`. Accessed: 2024-3-5.

[2] International bandwidth usage. `https://www.itu.int/itu-d/reports/statistics/2022/11/24/ff22-international-bandwidth-usage/`. Accessed: 2024-3-5.

[3] World record optical fiber transmission capacity doubles to 22.9 petabits per second. `https://www.nict.go.jp/en/press/2023/11/30-1.html`. Accessed: 2024-3-5.

[4] Jeff Hecht. Ultrafast fibre optics set new speed record. *New Sci.*, 210(2809):24, April 2011.

[5] Dan Marom, Paul Colbourne, Antonio D'Errico, Nicolas Fontaine, Yuichiro Ikuma, Roberto Proietti, Liangjia Zong, José Manuel Rivas-Moscoso, and Ioannis Tomkos. Survey of photonic switching architectures and technologies in support of spatially and spectrally flexible optical networking. *Journal of Optical Communications and Networking*, 9:1–26, 01 2017.

[6] George N. Rouskas and Harry G. Perros. *A Tutorial on Optical Networks*, page 155–193. Springer Berlin Heidelberg, 2002.

[7] Mingcong Yang, Qian Wu, Maiko Shigeno, and Yongbing Zhang. Hierarchical routing and resource assignment in spatial channel networks (scns): Oriented toward the massive sdm era. *Journal of Lightwave Technology*, 39(5):1255–1270, 2021.