July 2023

Multi-layer system capable of considering constraints and objectives from both IP and optical layers

Anonymous

Recommended Citation

Anonymous, "Multi-layer system capable of considering constraints and objectives from both IP and optical layers", Technical Disclosure Commons, (July 14, 2023)
https://www.tdcommons.org/dpubs_series/6056

Multi-layer system capable of considering constraints and objectives from both IP and optical layers

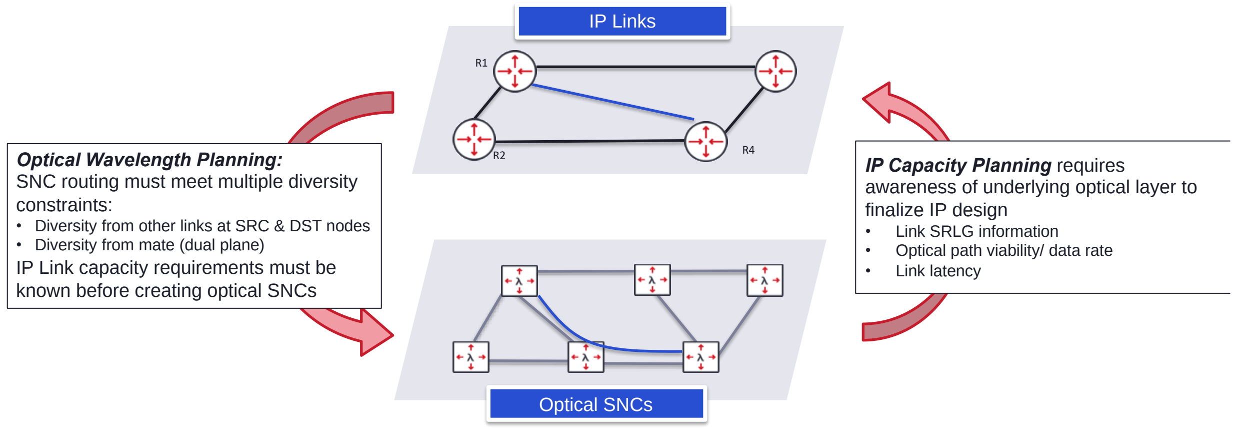
ABSTRACT

With networking trends driving traffic increasingly towards the edge of the network, IP-Optical convergence has come to the forefront of architecture evolution discussions. IP-optical convergence offers a means to streamline and simplify network operations and lower network costs. While network convergence has attracted a lot of industry interests in the areas of management and control and hardware integration, there has not been much focus on the amalgamation of the optical and IP layer design processes. At present, the IP and optical layers are planned by separate network teams with little or no co-ordination. The designs of each layer are made with limited cross-layer visibility, resulting in non-optimal solutions. The present disclosure proposes a system with full visibility of IP and optical layer design parameters.

DETAILED DESCRIPTION

The design of an IP network (IP and optical layers) requires many manual steps and is prone to errors since critical design parameters are manually transferred between the IP and optical teams using spreadsheets. Additionally, as Figure 1 illustrates, there are interdependencies in the design process of each layer resulting in the need to iterate the designs of each layer until a final solution is achieved. On one hand, knowledge of IP link capacities is required to determine the number of optical services required. On the other hand, IP topology design requires knowledge of optical layer routing and transport SRLG information to determine peak link bandwidth requirements under fault and optical path viability information (SNR) to determine if desired bit rates are achievable (ex. 400Gbps). This example highlights how the two layers are interconnected and that an iterative process is required to close the final design. The resulting solution is non-optimal in part because the x-layer information is incomplete and error prone, and because each layer is planned and optimized independently. Past internal research has shown that single layer optimization (ex. minimizing IP layer costs) can come at the expense of increased costs at the optical layer. Finally, coordinating this recursive design process between optical and IP engineers is a tedious process that can take up to several weeks. The goal is to develop a system that will automate the joint design process, remove the manual interventions and yield a dramatic reduction in time and improvements in the final product.

Today, IP and optical layers are designed separately. The design of an IP network (consisting of both IP and optical layers) is an iterative process in which IP and optical planning teams independently design and refine their respective network layer (Figure 2). Information is exchanged manually between teams as part of a feedback mechanism to finalize the design. Optical SNCs (the IP links) are individually, and sequentially routed by the optical planning tool. This tool has no intrinsic awareness of an IP topology, nor does it consider IP topology design objectives as part of the Optical SNC design process. Diversity constraints can be applied to an optical PCE to improve diversity, but route optimality is order dependent (impacted by previously routed links). Link routing is therefore not globally optimal from a diversity or path latency perspective.

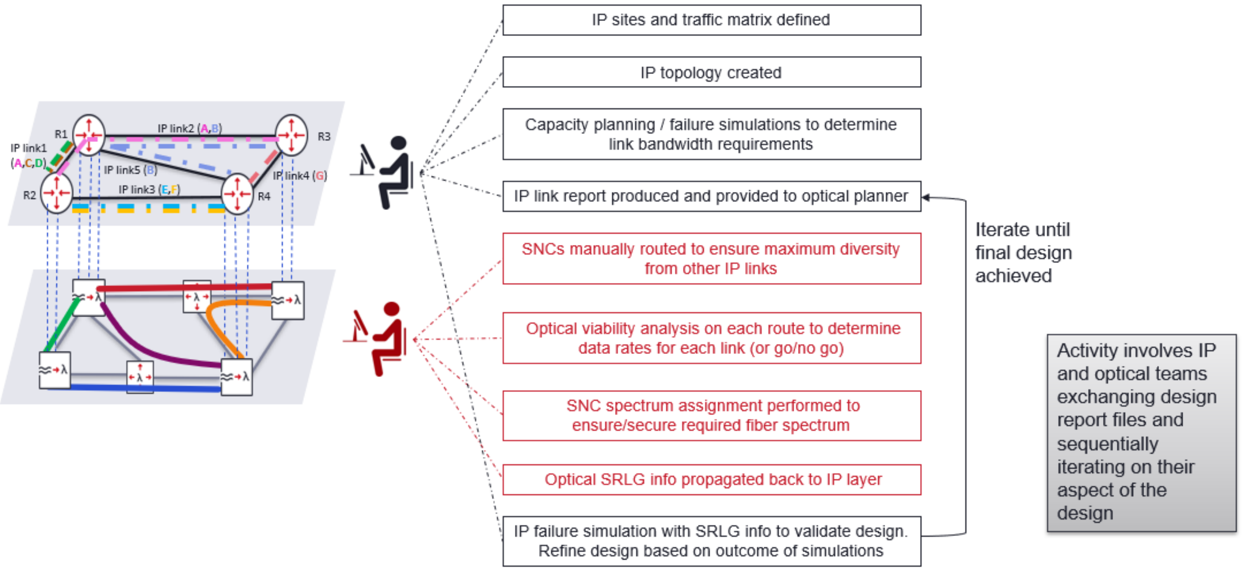


Optical Wavelength Planning:
SNC routing must meet multiple diversity constraints:
- Diversity from other links at SRC & DST nodes
- Diversity from mate (dual plane)

IP Link capacity requirements must be known before creating optical SNCs

IP Capacity Planning requires awareness of underlying optical layer to finalize IP design
- Link SRLG information
- Optical path viability/ data rate
- Link latency

Figure 1



IP sites and traffic matrix defined

IP topology created

Capacity planning / failure simulations to determine link bandwidth requirements

IP link report produced and provided to optical planner

SNCs manually routed to ensure maximum diversity from other IP links

Optical viability analysis on each route to determine data rates for each link (or go/no go)

SNC spectrum assignment performed to ensure/secure required fiber spectrum

Optical SRLG info propagated back to IP layer

IP failure simulation with SRLG info to validate design. Refine design based on outcome of simulations

Activity involves IP and optical teams exchanging design report files and sequentially iterating on their aspect of the design

Figure 2

Multi-Layer Optimization background:

This is a broad topic covering many distinct areas, some of which have been extensively covered in academic literature. Multi-layer optimization can refer to:

- **Network control:** Features in SDN controllers that improve the operational efficiency of a live network. ML optimization in this space could include:
  - Optically-aware IP path computation engines (ex. via propagation of optical SRLG info to IP PCE)

  - Strategies for reconfiguring IP and optical resources in response to faults
  - Dynamic reconfiguration of layers over time to re-optimize around shifting traffic patterns
- **Network Planning and Design:** Performed by offline tools Optimizations discussed in the context of network planning and design include:
  - Topology optimization – determination of most efficient way to interconnect nodes such as to minimize cost (resources). ML cost optimization would involve consideration of costs at both IP and Optical layers. It's believed that this is the most popular area of research regarding ML optimization.
  - Design of multi-layer resiliency strategies such as to reduce IP bandwidth requirements.

The present disclosure proposes a system with full visibility of IP and optical layer design parameters that:

- Integrates the IP and Optical layer design processes into one system to automate the recursive nature of the process and greatly reduce the design time and effort while also improving the quality of the design (Figure 3).
- Provides an optical path computation function for IP links that globally optimizes the placement of each link to create a topology that is optimally resilient to faults (maximal link diversity) and has lowest average link latency.

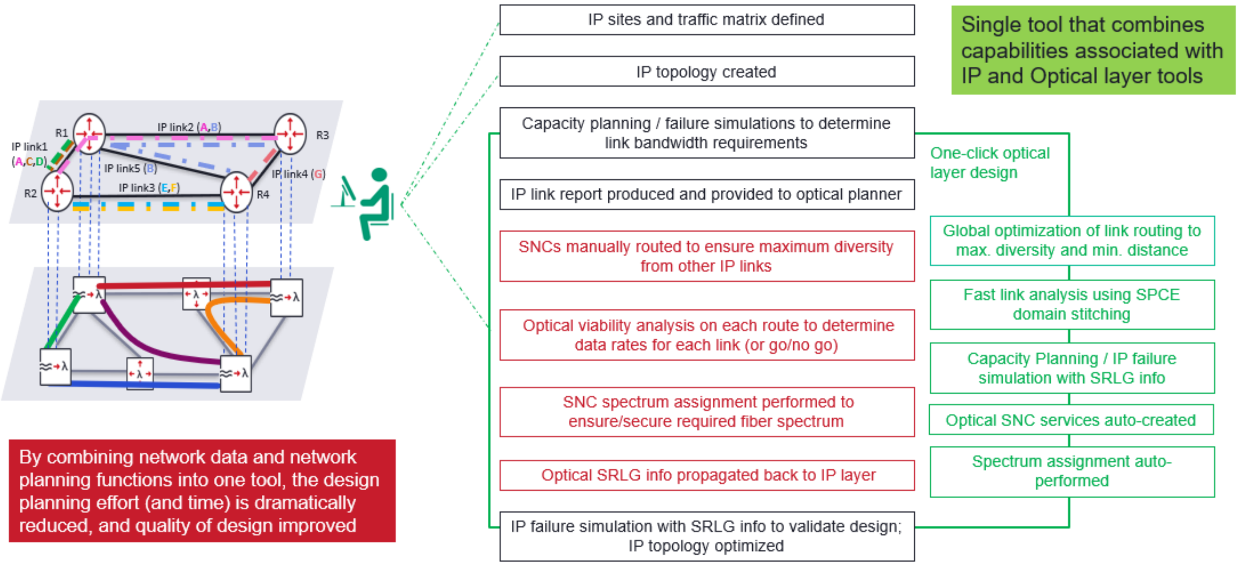


Figure 3

In the current mode of operations, satisfying the complicated diversity requirements between IP links cannot be handled in an automated fashion by optical layer. Diversity constraints are defined by IP link adjacencies and finding the most resilient IP topology requires to deal with highly tangled dependency graphs.

A global optimization engine looks at path placement options for all links, and through recursive analysis, determines the optimal routing that provides:

- Maximum diversity of links (delivering highest reliability)
- Lowest overall link length

The interdependencies between the layers are addressed by proposing a feedback-based iterative process. Each layer will interpret the received feedback as a set of constraints and take corrective actions to find the optimal multi-layer design. The proposed system has several use-cases and can handle both greenfield and brownfield designs. The followings are some of the use-cases proposed for this innovation.

- Flow through design process: automated instantiation of an optical layer design based on a pre-defined IP topology and traffic matrix
- IP Link diversity audit
- IP+DWDM greenfield design optimization
- IP+DWDM brownfield design optimization
- Multi-layer restoration simulation & design optimization
- IP+DWDM brownfield design augment

Figure 4 illustrates the mechanism for the proposed multi-layer system. The main inputs are photonic topology, IP traffic matrix and the placement of IP routers. If there is already an IP topology, the framework will use it and optimizes the current topology by proposing modifications e.g., adding/removing longer express links. For the problems where no IP topology is defined, the framework will start with an initial IP topology i.e., full mesh or hop-by-hop topologies and iteratively moves toward the optimized IP topology. A global route optimization engine (green box in Figure 4) assigns a physical route to each IP link. The designed routing scheme is a multi-objective population-based evolutionary algorithm. The fact that the optimization engine is population-based facilitates updating population by considering the constraints imposed by any of the two layers. For example, if a fiber link is congested and used on the shortest path by many IP links, the new population will enforce the load balancing by adjusting the weight of the fibers while calculating the new population. The routing optimization engine is also multi-objective and can handle several criteria e.g., link diversity requirement connected to each router in the same plane or mated links in different planes, latency, availability, etc.

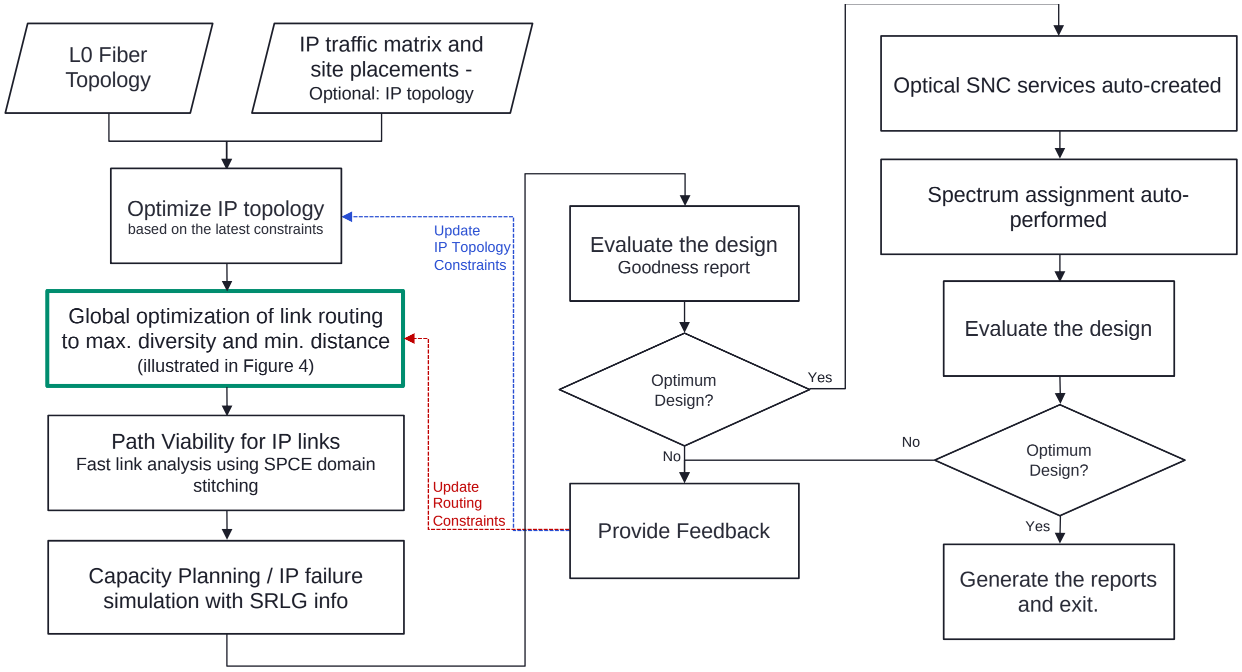


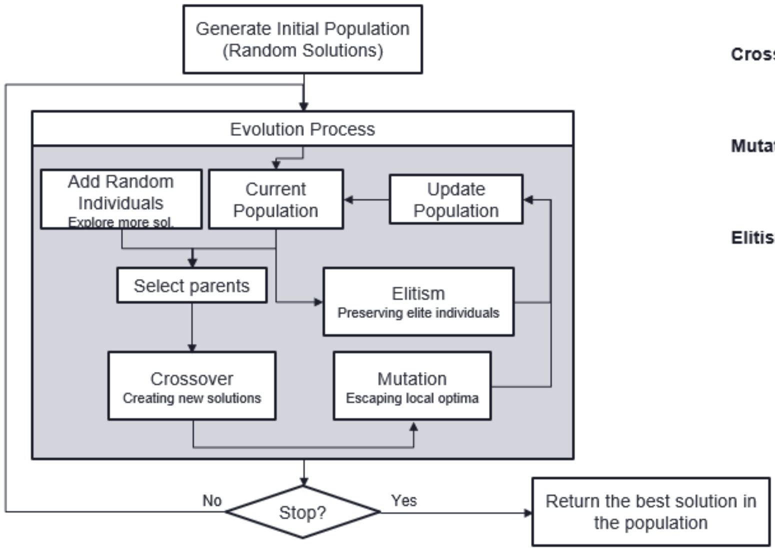
Figure 4

Figure 5 describes the iterative process designed inside the global route optimization engine (green box in Figure 4). The algorithm starts with a pre-defined set of solutions known as initial population. Every individual member of the population is a potential feasible solution to the routing problem i.e., satisfying the current set of constraints revealed while improving the design. The algorithm gradually identifies the most promising genes within the best individuals and creates improving solutions. The improving individuals enter the population and guide the algorithm toward the optimum solution. The algorithm searches the solution space and takes advantage of different embedded operations in order to maintaining the diversity of the population to prevent pre-mature convergence, escape from local optima and find the best solution.

Combining IP and optical layer network data and planning functionality into a single system and planning process to achieve:

- Higher quality designs, by aspects such as described in herein, and by the elimination of errors, data inconsistencies or incompleteness when information is transferred between IP and optical teams.
- Dramatic improvements in design planning times (multiple days/weeks to minutes).

Although various forms of multi-layer optimization have been discussed in the industry, there has been little or no discussion on the optimization of the planning process itself.



Crossover: the operator to combine the genetic information of two parents to generate new offspring

Mutation: the operator to maintain genetic diversity from one generation to the next

Elitism: carrying over the best organism(s) from the current generation to the next

Figure 5

The routing solution obtained from the route optimization engine provides the opportunity to check the viability of the paths and share the SRLG info with IP layer. Once the optimization engine returns the best photonic routing of IP links, the framework calls of a novel approach that evaluates the viability of the paths by stitching the domains and makes sure that the OSNR for the assigned path respects the threshold defined for the deployed technolog. This approach evaluates the viability of the paths by stitching the domains and makes sure that the OSNR for the assigned path respects the threshold defined for the deployed technology.

With the latest SRLG info received from L0, the framework runs the bundle failure analysis at IP layer and calculates the bandwidth requirement and consequently number of photonic services for all IP links.

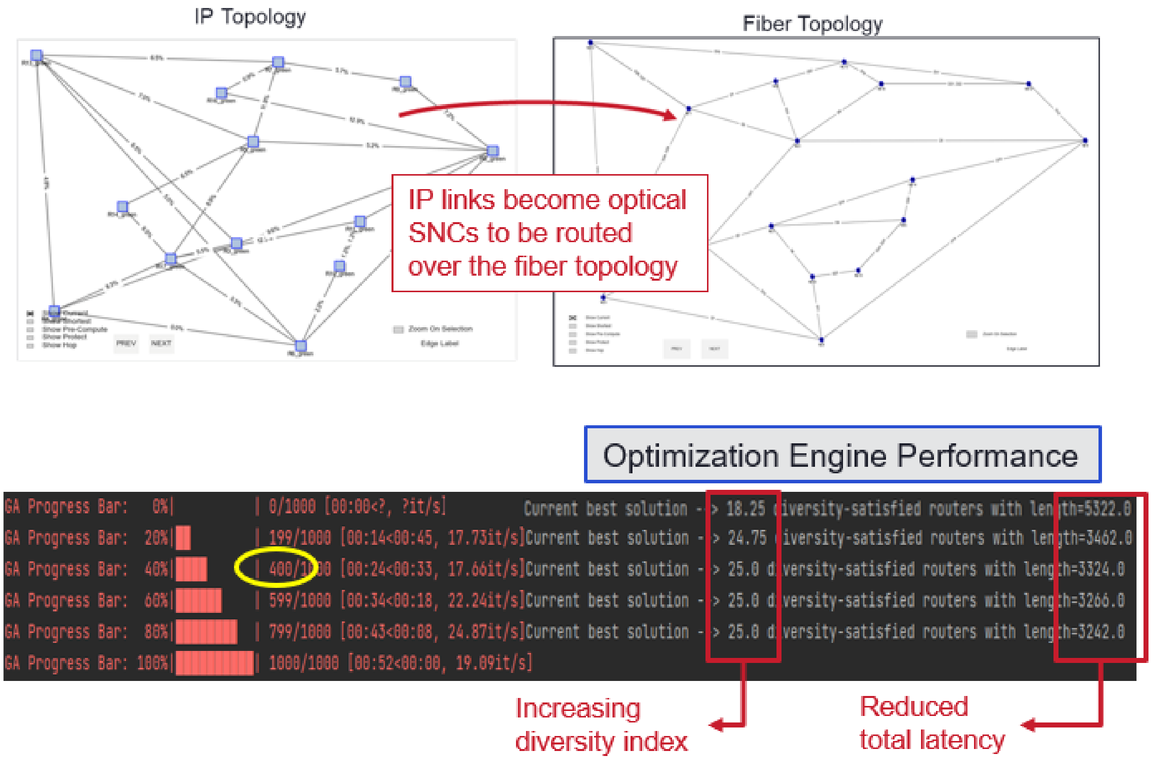
At this point, the tool will provide the goodness report, shown in Figure 8, on diversity between IP links, max achievable rates, load balancing on optical fibers, etc. If the current design is not optimum and violates the conditions set by the user, the tool will provide the feedback to the corresponding layers. For example, if there is a congested fiber link serving many optical SNC services, the solution could be discouraging some longer express links to improve the spectral efficiency or simply considering a different physical route assigned to some IP links to distribute the load more evenly. It is worth noting that changing the IP topology requires running the route optimization engine and changing the physical routes require updating the SRLG info and performing bundle failure analysis. In either case, both layers will receive the feedback, investigate the impact of the proposed changes and share the outcome with the other layer. Once the proposed design is satisfactory and meets the requirements, the framework auto-generates the optical SNCs and auto-performs the spectrum assignment.

The state of the optical layer is checked once the optical waves are routed. Fragmentation level, load balancing and service availability are among the investigated factors. Once again, the required modifications will be passed as the feedback to the right layer and corrective actions will be taken. The tool will stop and provide the results once all requirements are met. The goodness report along with suggested modifications/actions will be auto generated.

An initial version of the flow through process design was implemented. The performance of implemented framework is being studied on different instances. It has already been tested on two network captures

- Model A network: 68 IP routes and 164 IP links
- Model B network: 26 IP routers and 44 IP links

In both cases, the optimal design was found in less than 5 minutes. Figure 6 illustrates the algorithm running to globally optimize the diversity and average path length of IP links.



IP links become optical SNCs to be routed over the fiber topology

```
GA Progress Bar:   0%|          | 0/1000 [00:00<?, ?it/s]          Current best solution --> 18.25 diversity-satisfied routers with length=5322.0
GA Progress Bar:  20%|█         | 199/1000 [00:14<00:45, 17.73it/s]Current best solution --> 24.75 diversity-satisfied routers with length=3462.0
GA Progress Bar:  40%|███       | 400/1000 [00:24<00:33, 17.66it/s]Current best solution --> 25.0 diversity-satisfied routers with length=3324.0
GA Progress Bar:  60%|████      | 599/1000 [00:34<00:18, 22.24it/s]Current best solution --> 25.0 diversity-satisfied routers with length=3266.0
GA Progress Bar:  80%|█████     | 799/1000 [00:43<00:08, 24.87it/s]Current best solution --> 25.0 diversity-satisfied routers with length=3242.0
GA Progress Bar: 100%|██████████| 1000/1000 [00:52<00:00, 19.09it/s]
```

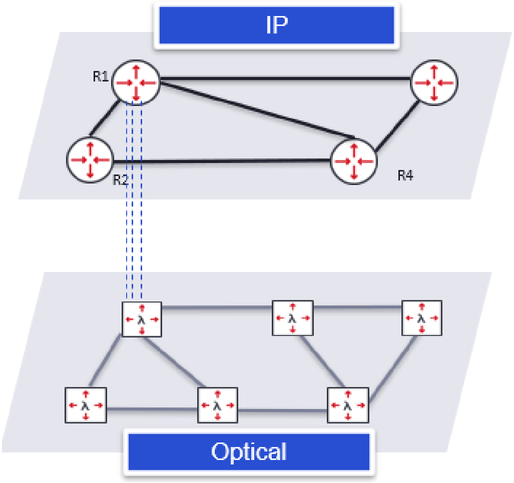
Increasing diversity index

Reduced total latency

Figure 6

Rather than designing/planning each optical SNC individually, the approach considers the placements of all IP links concurrently. The IP topology (and optical SNC routing) is globally optimized according to 2 metrics: link route diversity and path length. The implementation uses a genetic algorithm, but other optimization strategies could equally be applied.

How it works:
The system assigns a 'diversity target' score to each IP node based on fiber path diversity at that site and the number of IP links terminating at site. The path length of all IP links are summed to represent average path latency. The algorithm iterates to maximize diversity and minimize latency objectives.



# Diverse IP Links	Diversity Metric (R1)
1	1/3=0.33
2	2/3=0.66
3	3/3=1

Figure 7

Example referring to Figure 7:

A diversity metric is defined for each router.

- Number of diverse IP links

Router R1 has 3 IP links and is connected to a degree 3 ROADM.

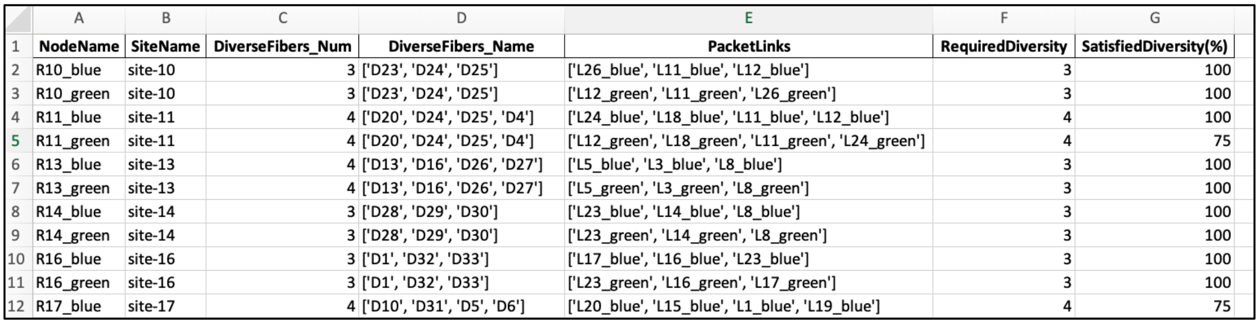
Ideally, systems will use all ROADM degrees and have all IP links diverse from each other.

- R1 will be able to communicate to other routers in the worst case

If all IP links are diverse, R1 will get value 1=3/3.

The goal is to maximize the diversity over all routers while minimizing overall link lengths.

Examples of goodness reports:



	A	B	C	D	E	F	G
1	**NodeName**	**SiteName**	**DiverseFibers_Num**	**DiverseFibers_Name**	**PacketLinks**	**RequiredDiversity**	**SatisfiedDiversity(%)**
2	R10_blue	site-10	3	['D23', 'D24', 'D25']	['L26_blue', 'L11_blue', 'L12_blue']	3	100
3	R10_green	site-10	3	['D23', 'D24', 'D25']	['L12_green', 'L11_green', 'L26_green']	3	100
4	R11_blue	site-11	4	['D20', 'D24', 'D25', 'D4']	['L24_blue', 'L18_blue', 'L11_blue', 'L12_blue']	4	100
5	R11_green	site-11	4	['D20', 'D24', 'D25', 'D4']	['L12_green', 'L18_green', 'L11_green', 'L24_green']	4	75
6	R13_blue	site-13	4	['D13', 'D16', 'D26', 'D27']	['L5_blue', 'L3_blue', 'L8_blue']	3	100
7	R13_green	site-13	4	['D13', 'D16', 'D26', 'D27']	['L5_green', 'L3_green', 'L8_green']	3	100
8	R14_blue	site-14	3	['D28', 'D29', 'D30']	['L23_blue', 'L14_blue', 'L8_blue']	3	100
9	R14_green	site-14	3	['D28', 'D29', 'D30']	['L23_green', 'L14_green', 'L8_green']	3	100
10	R16_blue	site-16	3	['D1', 'D32', 'D33']	['L17_blue', 'L16_blue', 'L23_blue']	3	100
11	R16_green	site-16	3	['D1', 'D32', 'D33']	['L23_green', 'L16_green', 'L17_green']	3	100
12	R17_blue	site-17	4	['D10', 'D31', 'D5', 'D6']	['L20_blue', 'L15_blue', 'L1_blue', 'L19_blue']	4	75

Figure 8

It will be appreciated that some embodiments described herein may include one or more generic or specialized processors ("one or more processors") such as microprocessors, digital signal processors, customized processors, and Field-Programmable Gate Arrays (FPGAs) and unique stored program instructions (including both software and firmware) that control the one or more processors to implement, in conjunction with certain non-processor circuits, some, most, or all of the functions of the methods and/or systems described herein. Alternatively, some or all functions may be implemented by a state machine that has no stored program instructions, or in one or more Application-Specific Integrated Circuits (ASICs), in which each function or some combinations of certain of the functions are implemented as custom logic. Of course, a combination of the aforementioned approaches may be used. Moreover, some embodiments may be implemented as a non-transitory computer-readable storage medium having computer-readable code stored thereon for programming a computer, server, appliance, device, etc. each of which may include a processor to perform methods as described and claimed herein. Examples of such computer-readable storage mediums include, but are not limited to, a hard disk, an optical storage device, a magnetic storage device, a ROM (Read Only Memory), a PROM (Programmable Read-Only Memory), an EPROM (Erasable Programmable Read-Only Memory), an EEPROM (Electrically Erasable Programmable Read-Only Memory), Flash memory, and the like. When stored in the non-transitory computer-readable medium, the software can include instructions executable by a processor that, in response to such execution, cause a processor or any other circuitry to perform a set of operations, steps, methods, processes, algorithms, etc.

Although the present disclosure has been illustrated and described herein with reference to preferred embodiments and specific examples thereof, it will be readily apparent to those of ordinary skill in the art that other embodiments and examples may perform similar functions and/or achieve like results. All such equivalent embodiments and examples are within the spirit and scope of the present disclosure.