ABSTRACT

It is much desired that the future DOE network possesses the ability to dynamically rearrange transport network topologies to enable the sharing of wide area transport capability among applications, and at the same time dynamically vary transport bandwidth to suit different high performance application needs. However, the unique network requirements to support DOE’s science mission result in challenging issues that are not addressed by existing networking techniques and methodologies.

We propose to study the space-time demand scheduling problem and design efficient algorithms to automate scheduling of resource demands (e.g., bandwidth) of applications. The resultant demand schedule, routes and wavelength assignments will then be used by the signaling protocol to configure the network. Since the speed of configuration determines routing and connection efficiency (e.g., long configuration times are especially not efficient for short sessions), we consider the efficient establishment and re-use of logical topologies in our algorithms so that the reconfiguration time is minimized to speed up provisioning. We will investigate the provisioning of multi-resolution on-demand bandwidths and efficient terabit traffic grooming in the space-time scheduling framework. We will also consider the need of multicast support in the future DOE network by proposing a dynamic multicast session provisioning algorithm to facilitate large distributed collaborative applications. These efforts will be a step towards enabling on-demand bandwidths by decoupling the service bandwidth from the infrastructure bandwidth. Finally, we will examine ways in which cost-effective survivability can be provided in mesh topology networks so that ring like restoration speed is attained while cost of spare resources is minimized. This provides quality protection options for the future DOE network should such needs are deemed necessary.

DESCRIPTION OF ACCOMPLISHMENTS

1. Space-Time Demand Scheduling

We are focused on dynamic service provisioning in DOE WDM optical networks. We envision, in the near future, that the scheduling of application resource demands (e.g., on-demand bandwidths) may be handled by a central server because of the relatively small scale of the DOE network. This central server based architecture may evolve. However, the algorithms developed can be adapted to work in a distributed environment. The server will run a scheduling algorithm that determines the route, wavelength assignment (i.e., spatial information), and the time interval during which a demand is to be accommodated. The network will then be configured through the signaling protocol based on the schedule determined by the server. The problem is termed as the space-time demand scheduling problem.

Specifically, given the network topology $G = (V,E)$, where $V$ is the set of nodes and $E$ is the set of links. Each link has $W$ wavelengths. A set of light-path demands $M$ is given, each of which is represented by a tuple $(s,d,n,\alpha,\beta,p)$ where $s$ is the source, $d$ is the destination, $n$ is the number of light-paths required (traffic grooming will be considered later), $\alpha$ and $\beta$ are the starting time and ending time of the demand, $p$ is the priority of the demand (for example, $p = 1$ indicates a high priority and $p = 0$ a low priority). A high
priority demand is one whose starting time and ending time are fixed and the demand should be accommodated during the given time interval while the time interval of a low priority demand is negotiable, and therefore can be moved forward or backward in time. The problem is to route each demand and assign a proper wavelength to each demand such that

- **In the non-blocking case** in which the network has enough resources to accommodate all the demands, the goal is to minimize the total network resources used in terms of
  
  1. the total number of wavelength links used by all demands; or
  2. the maximum number of wavelengths needed on a link to balance the network load.

- **In the blocking case** in which the network does not have enough resources to accommodate all the demands as specified, the goal is
  
  1. to minimize the number of demands to be rearranged (i.e., to minimize the subset of demands that may have to be moved forward or backward in time) in order to have all the demands in the set $M$ accommodated by the network; or
  2. to minimize the total time from the time when the first demand arrives to the time when the last demand ends (termed as the schedule length) it takes to satisfy all the demands in $M$.

**Resource Demand Model** A good model for on-demand bandwidths is very important. We initially use the demand model $(s, d, n, \alpha, \beta, p)$ to characterize the application resource requirements. This model is different from the simple dynamic traffic model commonly assumed in the literature. The model that is closest to ours is perhaps that of [6]. In our model, the time dimension of demands is explicitly considered since transactions in ultra high-speed networks will be short-lived in contrast to $7 \times 24$ operations. We believe that this model may be more suitable to characterize certain DOE network application resource requirements (e.g., non-IP dedicated connections). Note that the model may allow the applications to negotiate their starting times and ending times with the scheduler. The validity of this model is subject to whether or not the parameters (i.e., starting time, ending time, bandwidth requirement etc) can be provided by the applications. The availability and accuracy of the network state information and accessibility by the central server are also issues to consider when the scheduling algorithm is to be deployed. The model will be refined and modified as the PI interacts with the DOE application community and DOE UltraNet staff. For example, a new model may allow a range for starting times and ending times, or allow multiple split time intervals during which an application may require resources and so on.

The optimization of space-time demand scheduling problems with different objectives are formulated and given in Appendix B. The solutions to the optimization problems serve as the baseline for comparing proposed space-time demand scheduling algorithms. We have designed scheduling algorithms to solve the space-time demand scheduling problem and its variants.

**1.1 Time Window Based Scheduling Algorithm**

The hardness of the problem lies in the spatial and temporal constraints imposed on the set of demands. In the spatial domain, demands are routed through the mesh network topology and may share the same wavelength on the same link. In the temporal domain, demands may overlap in time. We propose heuristic algorithms that take advantage of resource reuse in both domains.
Table 1: An example of a scheduled demand set

<table>
<thead>
<tr>
<th>demand</th>
<th>s</th>
<th>d</th>
<th>n</th>
<th>α</th>
<th>β</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>r₁</td>
<td>B</td>
<td>F</td>
<td>1</td>
<td>05:00</td>
<td>09:20</td>
<td>0 (low)</td>
</tr>
<tr>
<td>r₂</td>
<td>B</td>
<td>D</td>
<td>1</td>
<td>07:00</td>
<td>12:40</td>
<td>0 (low)</td>
</tr>
<tr>
<td>r₃</td>
<td>A</td>
<td>C</td>
<td>2</td>
<td>08:00</td>
<td>14:00</td>
<td>1 (high)</td>
</tr>
<tr>
<td>r₄</td>
<td>A</td>
<td>D</td>
<td>2</td>
<td>11:00</td>
<td>16:00</td>
<td>0 (low)</td>
</tr>
<tr>
<td>r₅</td>
<td>E</td>
<td>D</td>
<td>3</td>
<td>12:00</td>
<td>14:50</td>
<td>0 (low)</td>
</tr>
<tr>
<td>r₆</td>
<td>C</td>
<td>E</td>
<td>1</td>
<td>17:00</td>
<td>21:00</td>
<td>1 (high)</td>
</tr>
<tr>
<td>r₇</td>
<td>F</td>
<td>A</td>
<td>3</td>
<td>18:00</td>
<td>21:00</td>
<td>0 (low)</td>
</tr>
</tbody>
</table>

Our algorithm is based on the following observations. Demands that overlap in time must be disjoint in the spatial domain (e.g., they cannot share the same wavelength on the same link). Network resources used by one demand can be reused by another demand as long as they do not overlap in time. This motivates us to divide the set of demands into subsets based on the demands’ starting and ending times such that demands in different subsets are disjoint in time. We represent the set of demands \( M \) using an interval graph (e.g., Fig. 2)(b) where an interval represents the starting and ending times of a demand. Based on the interval graph representation, we can then divide the demands into subsets called time windows (Fig. 2)(b). An algorithm to divide demands into time windows is given later. A virtual network topology is associated with each time window. Note that it is not always possible to assign a demand into a single time window. This demand is termed as a straddling demand, e.g., \( r₂ \) and \( r₃ \) of Fig. 2(b). The resources (e.g., wavelength-links) used by demands in one time window can be reused by the demands in other windows except those demands that straddle these time windows. Therefore, for straddling demands the same resources have to be reserved across time windows. In the following, we first describe the details of dividing the demand set into time windows and the algorithm for scheduling all the demands based on these windows. We then give the algorithm for solving the entire space-time demand scheduling problem.

**Time Window Division** The algorithm divides the set of demands into disjoint time windows. Demands within a time window overlap in time pairwise. Therefore, all demands within the same window have to be disjoint in the spatial domain, i.e., they cannot share the same wavelength on the same link. Specifically, let

\[
T = \bigcup_{i=1}^{M} \{ \beta_i \}
\]  

be an ordered set of \( |T| \) demand ending time values \( T_1 < T_2 < \ldots < T_{|T|} \) (\(|T| \leq M\) since some \( \beta_i \)'s may be the same). We take the values in \( T \) as possible division points. We define that a demand lies in a time interval if its starting time or ending time lies in the interval. We then adapt the maximum independent set algorithm over an interval graph [5] to find a maximum time interval (starting from the ending time of the previous time window) during which all demands overlap in time pairwise. The algorithm pseudo code is given in Fig. 1. Figure 2(b) shows the time window division result of an example demand set shown in Table 1 which has seven demands in an example network shown in Fig. 2(a). In particular, \( r₂ \) and \( r₃ \) are straddling demands.

After the time window division, a demand may reside within a certain time window or straddle two or more windows. In the latter case, the demand is accommodated only when the same path can be found and resources on the path can be reserved in all the virtual networks associated with all the time windows it straddles. Otherwise the demand has to be rearranged so that sufficient network resources can be provided in the new windows it straddles.

**Routing and Wavelength Assignment (RWA) Based on Time Windows** After dividing the original demand set using the time window division algorithm, we have a number of demand subsets: \( D^{SH} \), \( D^{SL} \), \( TW_i^{H} \), \( TW_i^{L} \).
Time Window Division Algorithm (M)

Begin

\[ T = \text{the set of possible division points}; \]
\[ t_0 = T_1; t_1 = T_2; W = \emptyset; \]
for \( i = 1 \) to \( |T| \)

\[ t_2 = T_i; \]
find all demands in \([t_0,t_2]\) and put them in \( W \);
find the cardinality of the maximum independent set of \( W \), \( W^* \);
if \( (W^* > 1) \) then

a new time window \( TW[t_0,t_1] \) and demands in it are determined;
\[ t_0 = t_1; W = \emptyset; \]
endif

\[ t_1 = t_2; \]
endfor

End

Figure 1: Pseudo code for time window division algorithm.

where \( D^{SH} \) and \( D^{SL} \) are the sets of high-priority straddling demands and low-priority straddling demands, respectively; \( TW_i^H \) and \( TW_i^L \) are the sets of high-priority and low-priority demands within time window \( i \), respectively. Table 2 shows the subsets after the time window division of the example shown in Fig. 2(b).

In addition to the above subsets, we keep a virtual network \( G_i \) for each time window \( TW_i \) to record its residual network resources. For each non-straddling demand \( \in TW_i \), we apply a modified Dijkstra’s algorithm to the wavelength graph of the network to find the best path for the demand in the virtual network of time window \( TW_i \). By best path, we mean that the number of wavelengths available on all the links along the path satisfies the capacity requirement of the demand and the path has the minimum total cost (in terms of e.g., path length). Once such a path is found for the demand, we update the network status, i.e., remove all the network resources used by the path from the virtual network of time window \( TW_i \). If the demand cannot be satisfied in its time window, the next procedure depends on the priority of the demand. If its priority is high (i.e., the demand belongs to some \( TW_i^H \)), the RWA algorithm stops because a high-priority demand could not be satisfied or this demand can be demoted to \( TW_i^L \) and given a preferential treatment within \( TW_i^L \); otherwise (i.e., the demand belongs to \( TW_i^L \)), we put the demand into demand subset \( D^R \) in which all the demands need to be rearranged (Fig. 3).

For straddling demands (in \( D^{SH} \) or \( D^{SL} \), we handle them in the same way except that the virtual network in which each demand is routed is the intersection of all the virtual networks of the time windows straddled by the demand, and all these virtual networks must be updated if a path is found. For each demand

<table>
<thead>
<tr>
<th>( TW^H )</th>
<th>( Window1 )</th>
<th>( Window2 )</th>
<th>( Window3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_1 )</td>
<td>( r_4,r_5 )</td>
<td>( r_7 )</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Subsets after time window division
subset \((D^{SH} \text{ and } D^{SL})\), in addition, we select a demand for routing and wavelength assignment in the descending order of request capacity in the set. Our objective is to accommodate as many demands with higher capacity requirements as possible first. Therefore, this algorithm, in this sense, is a greedy RWA algorithm. The pseudo code of the algorithm that handles \(TW^H_i\), \(TW^L_i\), \(D^{SH}\), and \(D^{SL}\) is given in Fig. 3.

**Greedy Time Window RWA Algorithm (G, R)**

Begin // R: the set of demands to be scheduled
while \((R \neq \emptyset)\) do
    choose demand \(r\) which has the highest capacity requirement in \(R\);
    \(G' = G\);
    for each time window \(TW_i\) straddled by demand \(r\)
        \(G' = G' \cap G_i\);
    endfor
    find the shortest path \(p_r\) for demand \(r\) in \(G'\);
    if \(p_r\) is found
        \(R = R - r\);
        for each time window \(TW_i\) straddled by demand \(r\)
            \(G_i = G_i - p_r\);
        endfor
    else
        if the priority of demand \(r\) is high
            demote \(r\);
        else
            put \(r\) in \(D^R\);
        endif
    endif
endwhile
End

Figure 3: Greedy time window based RWA algorithm.

If the network resources are not sufficient to satisfy all the demands as specified in their original requests, some of the low-priority demands have to be rearranged. That is, some of the low-priority demands can be accommodated in terms of their capacity requirements at the cost of changed schedules. \(D^R\) is the demand set in which all the demands must be rearranged. For each demand in \(D^R\), we try to find one or more time windows in which the demand can be accommodated and the window(s) starts/start as early as possible. The rationale is that we prefer to schedule the demand by reusing resources and we would like to prevent the rearrangement of
Rearrange RWA Algorithm (G, R)
Begin //TW_i has an associated virtual network G_i
while (R ≠ ∅) do
    choose demand r which has the highest capacity requirement in R;
    pr = ∅; i = 1; // pr, route and wavelength assignment for r
    while (pr = ∅) do
        find the most right time window TW_j such that windows i to j can accommodate demand r;
        if no such time window is found
            create a new time window TW_j with G_j for demand r;
        else
            G' = G_i ∩ ... ∩ G_j;
        endif
        find the shortest path pr for demand r in G';
        if pr is found
            R = R – r;
            for each time window TW_i straddled by demand r
                G_i = G_i – p_r;
            endfor
        else
            i = i + 1;
        endif
    endwhile
End

Figure 4: RWA algorithm for rearranged demands.

the demand from prolonging the total time it takes to schedule all demands in M (also termed as the schedule length). The pseudo code of the RWA algorithm for rearranged demands is given in Fig. 4.

Space-Time Demand Scheduling Algorithm
In the space-time demand scheduling problem, our objective is to accommodate all high-priority demands and as many low-priority demands as possible with both their capacity requirements and schedule requirements. If not all the demands can be satisfied in both space and time domains at the same time, some low-priority demands must be selected as victims to be rearranged. Therefore, to schedule the demand set M, we try to route and assignment wavelengths to high-priority demands first, and then solve the RWA problem for low-priority demands. In addition, straddling demands are routed prior to the demands residing in a single time window since it is more likely to route a straddling demand earlier. The pseudo code of the overall time window based RWA algorithm is given in Fig. 5.

Time Space Demand Scheduling Algorithm (G, M)
Begin
    run Time Window Division Algorithm (M);
    run Greedy Time Window RWA Algorithm (G, DSH);
    run Greedy Time Window RWA Algorithm (G, TW_i^H) for all time window TW_i;
    run Greedy Time Window RWA Algorithm (G, DSL);
    run Greedy Time Window RWA Algorithm (G, TW_i^L) for all time window TW_i;
    if (DR ≠ ∅), run Rearrange RWA Algorithm (DR);
End

Figure 5: Space-time demand scheduling algorithm.
1.2 Traffic Matrix Based Scheduling Algorithm

In this algorithm, we divide the demand set \( M \) into a number of subsets using a similar approach used by the time window based scheduling algorithm. The difference is that demands in a subset do not have to be disjoint in time pairwise. We then construct a traffic matrix for each demand subset. A traffic matrix characterizes the traffic between source-destination node pairs. The demand set \( M \) can then be considered as time varying traffic that is modeled by a set of traffic matrices. A framework was proposed in [18] to solve the RWA problem when the offered traffic is characterized by a set of traffic matrices - a variant of dynamically changing traffic. We observe that the demand matrices we obtain can be regarded as the traffic matrices of [18]. We then adapt the algorithm proposed in [18] to solve our space-time demand scheduling problem. More specifically, the adapted algorithm needs to handle priorities of demands, straddling demands (demands that reside in multiple traffic matrices), and rearrangement of demands.

1.3 Performance Evaluation

We are currently performing simulation studies to evaluate the performance of the proposed algorithms and to enhance the algorithms. We are looking at time complexity of the proposed algorithms, algorithm performance with respect to demand time correlation (i.e., to what extent demands are correlated in time), the tradeoff between demand time correlation and time window selection in the traffic matrix based scheduling algorithm, and so on. As the on-demand bandwidths may vary for different applications, we plan to include traffic grooming (i.e., to efficiently deal with multi-resolution on-demand bandwidths) in the space-time scheduling framework. Finally, we also plan to adapt the proposed algorithms to handle dynamically arrived demands (i.e., demands are not given by a known set \( M \); rather, a demand dynamically arrives with a requested lasting time) using a moving time window based scheduling approach.

The performance results and fine-tuned algorithms will be reported in the next progress report.

2. Dynamic Multicast Session Provisioning in WDM Optical Networks

Large, distributed collaborative projects are increasingly common in DOE. Current IP-multicast is still a fragile technology. While efforts on making IP-multicast more robust are on-going, we believe that as the DOE network migrates to WDM based optical networks, support of multicast within the optical network becomes necessary for many reasons, for example, high bandwidth requirements, guaranteed performance, and effective sharing of resources with other types of services (e.g., dedicated non-IP service).

In order to effectively support multicast, nodes in a WDM network need to have additional capabilities than typical WDM nodes have. Nodes may have the capability of tapping a small amount of optical power from a wavelength signal and forwarding the rest to appropriate nodes. This capability is termed as tap-and-continue (TaC). In this case, multicast can be achieved through one or more light-paths that include all the nodes in the multicast group. Each light-path requires a separate transmission from the source. This approach is characterized by the light-path model [13]. On the other hand, if a node is equipped with light splitters (called a splitting and delivery node (SaD node)), it can split the optical power of a wavelength signal into multiple parts and route them to multiple outgoing links. Each transmission from the source results in a light-tree. This approach is characterized by the light-tree model [14]. With all the nodes having both splitting and wavelength conversion capability, a single light-tree may be constructed for a multicast session. In networks with sparse splitting capability (i.e., to reduce cost), it may not be possible to include all the nodes of a multicast group in a single multicast tree, thereby calling for the construction of a set of light-trees. The set of light-trees required
for a single multicast session is called a light-forest. Each light-tree requires one separate transmission using a different wavelength on the same link, or may use the same wavelength on a different link, or a different wavelength on a different link.

In this work [19], we consider dynamic multicast session provisioning in a futuristic scenario in the DOE network. We assume the wavelength-routed WDM network with an arbitrary mesh topology which is represented by a graph \( G = (V, E) \) where \( V \) is the set of nodes, and \( E \) is the set of unidirectional edges. The network consists of nodes with full, partial, or no wavelength conversion capabilities. We assume that splitting and delivery (SaD) nodes are available and have full splitting capability. An SaD node is able to perform \( n \)-way light splitting. This type of devices is becoming commercially available. In our continuing work, we will consider the case where no SaD nodes are available and space-time scheduling of multicast sessions. We study the provisioning of dynamic multicast sessions in WDM optical networks with sparse splitting capability (that is, only a selected set of nodes is equipped with splitting capability) with the objective of minimizing network resources used.

The problem of dynamic multicast session provisioning in WDM optical mesh networks is to route the multicast traffic session from each source to the members of every multicast group, and to assign an appropriate wavelength to the session. Traditionally, routing and wavelength assignment are considered as two separate problems. To optimally provision multicast sessions, optimizing each problem separately and then combining them together, in general, do not yield an optimal solution to the complete problem. To derive the optimal solution, routing and wavelength assignment need to be considered simultaneously. We therefore formulate the problem as an integer linear program (ILP). The ILP formulation is general in that it only assumes that the network has partial light splitting capability.

We propose and evaluate an efficient heuristic algorithm for routing and assigning wavelengths to dynamically arrived multicast session requests. The solution to the ILP formulation serves as the baseline for evaluating the performance of the proposed algorithm. The basic idea of the proposed dynamic session provisioning algorithm is to first find a set of minimum paths (in terms of number of hops) from the source node to the destination nodes. Upon finding such a set, the rest of the algorithm tries to merge these paths to form a light-tree or a set of light-trees (a light-forest) that span to reach all the destination nodes. The minimum hop paths are found using the Dijkstra’s algorithm. The algorithm tries to make full use of SaD nodes in the network. Note that different SaD node assignment policies can be used with our proposed algorithm.

We have evaluated the effectiveness of the proposed online algorithm via simulation in terms of the connection blocking probability and the resources used by a session. All the simulation results are based on an average of 20 simulation runs. The 14-node NSFnet topology (Fig. 8(a)) is used in the simulation with one pair of unidirectional fibers between two directly connected nodes. The number of wavelengths supported on each fiber is 8. Nodes in the network are randomly assigned to be equipped with full range wavelength converters and the other nodes have partial wavelength conversion capabilities. Multicast session demands (e.g., the source node and the destination nodes) are randomly generated.

The first set of experiments were conducted to check for efficiency of the heuristic dynamic session provisioning algorithm. Fig 6(a) shows the results where average session lasting time is set at 1 time unit (with exponential distribution), and network nodes are randomly assigned as SaD nodes in this set of experiments. As seen from Fig. 6(a), the curves show a general tendency of growth in the percentage of sessions routed in the network as the total number of SaD nodes increases. The proposed dynamic session provisioning algorithm performs within 10-20% of the ILP solutions. The growth rate is higher for the heuristic algorithm, indicating that the algorithm performs better when the number of SaD nodes becomes larger.

Fig. 6(b) depicts the results from a set of experiments similar to the first set. Rather than randomly
assigning SaD nodes, the splitting capable nodes are assigned to nodes based on the nodes’ frequency of use during multicast routing. By nodes’ frequency of use during multicast routing, we mean that the node most frequently used in multicast session routing while there are no splitting capable nodes in the network is chosen first to be a SaD node, then the next frequently used node is chosen, and so on. The results of Fig. 6(b) show that for the ILP solution, this SaD node assignment policy does not improve the performance as much compared with the previous set of experiments. However, the heuristic algorithm, in this situation, performs much better when only a few SaD nodes are available in the network.

Fig. 7(a) illustrates the results of our next set of experiments conducted. We measure the blocking probability of multicast sessions as more SaD nodes are added to the network with varying traffic load. Splitting capable nodes in the network are randomly assigned. As seen from Fig. 7(a), the multicast session blocking probabilities for both the ILP solutions and the heuristic algorithm increase as traffic load increases. We also observe that the heuristic algorithm performs better and benefits more from the added splitting capable nodes than the ILP.

The last set of experiments look at the size of routing trees in terms of number of links used. In Fig 7(b), the average number of links used by a multicast session obtained by the ILP and that by the heuristic algorithm are computed as a ratio. The figure shows that the average number of links used per multicast session in the heuristic algorithm increases until about 50% of nodes in the network are SaD nodes. Thereafter, as more nodes are SaD nodes, the curves remain very close. This shows that the multicast trees constructed by our heuristic algorithm is very efficient when only a fraction of the nodes are SaD nodes. The results and findings have been submitted for publication to IEEE Globecom, 2004.

3. Optimal Configuration of $p$-Cycles in WDM Optical Networks with Partial Wavelength Conversion

In the future DOE network, delivery of terabytes of data per transaction will be the norm rather than the exception. Since a failure in a WDM optical network may result in a tremendous amount of data loss, efficient protection of data transport is thus very important. There are two types of well-known approaches to protecting optical networks: ring-based protection [3] and path-based protection [21, 12, 20]. Ring-based approaches are
Figure 7: (a) The session blocking probability versus different session loads and number of SaD nodes. (b) Ratio of the average number of links used for establishing a session in the heuristic algorithm over that of solutions obtained by ILP.

easier to manage and offer very fast protection switching. However, they are rather capacity inefficient. Mesh-based approaches, on the other hand, require much less spare capacity, but have the drawback of complicated signaling and network management for protection. The pre-configured protection cycle (p-cycle) based techniques [4] combine the benefits of both ring-based and path-based approaches in that it can achieve ring-like recovery speed while retaining the desired capacity efficiency of path-based protection approaches. Note that cost-effective fast restoration is very much desirable for the DOE production network. This effort provides quality protection options for the future DOE network should such needs be deemed necessary.

When wavelength conversion is not available, a light-path must use the same wavelength on all the links traversed in a WDM optical network. This requirement is known as the wavelength continuity constraint. On the other hand, if wavelength routers are capable of wavelength conversion, an optical signal may be converted from one wavelength to another wavelength. In some previous work on p-cycle based protection [15, 17, 16], a light-path following the wavelength continuity constraint is called a wavelength path (WP) while a virtual wavelength path (VWP) is defined to be a light-path that uses wavelength converters at each node on the path and may have different wavelengths on different links that the path traverses. Therefore, a WP network has no wavelength conversion capabilities at all while a VWP network has full wavelength conversion at every node, i.e., there are sufficient converters at each node to convert any incoming wavelength to any outgoing wavelength.

In this work [8], we study the optimal configuration of p-cycles in survivable WDM optical mesh networks with partial wavelength conversion while 100% restorability is guaranteed against any single failures. We formulate the problem as two integer linear programs for non-joint and joint optimization cases, respectively. In the non-joint approach, working paths are known before protection configuration is processed. p-cycles and wavelength converters are then optimally determined. In the joint approach, working paths, p-cycles, and wavelength converters are jointly determined. The objectives in both cases are to minimize the total cost of link capacity used by working paths and p-cycles as well as the cost of wavelength converters required to accommodate a set of traffic demand. In the proposed p-cycle configuration architectures, we take into account converter sharing: (a) when converters are used for accessing p-cycles and for wavelength conversion between two adjacent on-cycle spans; (b) when converters are used among disjoint straddling spans incident to the same node for accessing p-cycles. Converter sharing reduces network cost by requiring as few converters as possible. Our
simulation results indicate that the performances of the proposed approaches outperform existing approaches in terms of total network cost and total number of converters required. In addition, our approaches result in more balanced converter use in the network.

Specifically, we evaluate the performance of our proposed optimization models against that of the approaches for \( VW P \) networks and for networks in which converters can only be used to access \( WP \) \(-\)cycles from \( WP \) working paths [17, 16]. Two cases, non-joint optimization and joint optimizations, are considered in each of the three approaches. Therefore, overall six architectures are investigated in the simulation and abbreviated as follows:

- \( P W P \_N J \): working paths use \( WP \); partial wavelength conversion on \( p \)-cycles if required; non-joint optimization; (our approach)
- \( P W P \_J \): working paths use \( WP \); partial wavelength conversion on \( p \)-cycles if required; joint optimization; (our approach)
- \( W P \_N J \): working paths use \( WP \); protection \( p \)-cycles use \( WP \); \( p \)-cycles are accessed with wavelength conversion if required; non-joint optimization
- \( W P \_J \): working paths use \( WP \); protection \( p \)-cycles use \( WP \); \( p \)-cycles are accessed with wavelength conversion if required; joint optimization
- \( V W P \_N J \): working paths use \( WP \); protection \( p \)-cycles use \( VW P \); non-joint optimization
- \( V W P \_J \): working paths use \( WP \), protection \( p \)-cycles use \( VW P \); joint optimization

In the non-joint optimizations (i.e., \( P W P \_N J \), \( W P \_N J \) and \( V W P \_N J \)), we use the shortest path routing algorithm to find a working path with the shortest total link length for each demand, and apply the First-Fit strategy to assign required number of wavelengths to the path. For the joint optimization cases (i.e., \( P W P \_J \), \( W P \_J \) and \( V W P \_J \)), on the other hand, we take all shortest routes as the set of eligible routes. In both cases, each working path is a \( WP \) (i.e., the same wavelength is assigned to all spans along the path) and a set of candidate \( p \)-cycles is given as input for optimal \( p \)-cycle configuration.

Two example networks shown in Fig. 8 are used for performance evaluation and comparison. We first evaluate the performance of three non-joint approaches using the NSFNET topology (Fig. 8(a)). Due to the relatively high computational complexity of the joint methods, especially for \( P W P \_J \) and \( W P \_J \), we choose to compare the performance of the six architectures using a smaller 10-node network shown in Fig. 8(b). We
use demand sets containing 40 and 20 bidirectional pairs for the NSFNET topology and the 10-node network, respectively. The source and destination of traffic requests are randomly generated. In addition, we assume two fibers per span and 20 wavelengths per fiber in the NSFNET topology while the 10-node network has 10 wavelengths per fiber. Moreover, the maximum number of converters placed at each node is fixed to 40 and 30 in the two networks, respectively. However, the WP and VWP approaches do not follow this constraint, i.e., the number of converters used at each node is unlimited in the WP and VWP approaches.

**Total Cost Comparison** The total cost required to accommodate the traffic demands includes the cost of wavelength converters and link capacity costs incurred by all working paths and p-cycles. Fig. 9(a) shows the total costs required by the three non-joint approaches, respectively, as the cost of one wavelength converter is varied relative to the cost of one wavelength-link in the NSFNET topology. We express the converter cost in terms of the cost of a wavelength-link of certain length in order to figure in the factor that wavelength conversion cost may vary greatly with different technologies and advancement of technologies over time. From the figure, we observe that our non-joint approach (PWP\textsubscript{NJ}) incurs much less cost than the other two approaches. When the converter cost is 60 units, for example, the total cost incurred in our approach (PWP\textsubscript{NJ}) is only about 77% and 70% of that in WP\textsubscript{NJ} and VWP\textsubscript{NJ} approaches, respectively.

Fig. 9(b) shows that the total costs of our proposed optimization models are much less than those in VWP networks in both the non-joint and joint cases (i.e., VWP\textsubscript{NJ} and VWP\textsubscript{J}), and even less than the non-joint WP approach in which wavelength conversion is only used for WP working paths to access WP p-cycles (i.e., WP\textsubscript{NJ} case). The main reason is that a smaller number of converters is used in our approaches. Although the total cost required by WP\textsubscript{J} is close to that of PWP\textsubscript{J}, we observe that WP\textsubscript{J} needs more wavelengths on each span to satisfy all demands. In other words, a set of demands which can be accommodated by PWP\textsubscript{J} may not necessarily be satisfied by WP\textsubscript{J} if the number of wavelengths on each span in the network is not large enough. We also observe from the figure that the performance of joint optimizations in different architectures perform much better than its corresponding non-joint counterparts as reported in [17, 16].

**Impact of Converters** Our approaches deploy working paths and p-cycles so that the total cost of link capacities and wavelength converters used in the entire network is minimized to accommodate a set of traffic demands. Our optimization models take converter sharing into consideration to reduce costs by requiring as few converters as possible. In our approaches, in case of a failure, the converters at a node can be used either for working paths to access p-cycles when wavelength conversion is necessary or for wavelength conversion between two adjacent links on the same p-cycle. The converters can also be shared among disjoint straddling spans incident
to the same node for working paths to access \( p \)-cycles. In contrast, converter sharing is not considered at all in [17, 16]. In their partial conversion approach, converters can only be used for working paths to access \( p \)-cycles. In their \( VWP \) case, on the other hand, converters are used exclusively for working paths to access \( p \)-cycles or for wavelength conversion between two adjacent on-cycle spans. At the common node to which straddling spans are incident, moreover, a certain number of dedicated converters are needed for each straddling span. Therefore, their approaches need much more converters to accommodate all traffic demands as shown in Fig. 10. The results indicate that our approaches in general perform much better than other approaches except that \( PW P J \) slightly outperforms \( WP J \) as shown in Fig. 10(b). For example, when the converter cost is 40 units, \( PW P N J \) uses about 42% and 30% converters compared with \( WP N J \) and \( VWP N J \) approaches, respectively, in Fig. 10(a) while \( PW P J \) uses about 50% and 2% converters compared with \( WP J \) and \( VWP J \) approaches, respectively, in Fig. 10(b). However, \( WP J \) needs more wavelengths on each span to accommodate all traffic demands.

In addition to the total number of converters used, we also consider the maximum number of converters required among all the nodes in the network when evaluating the performances of different approaches. This quantity indicates the fairness of different approaches in converter requirement and use. A smaller number implies that the difference in the numbers of converters used at various nodes may be smaller. Therefore, converter use may be more balanced in the network. Fig. 11 shows that our approaches require a much smaller maximum number of converters among all the nodes and therefore outperform other existing approaches. When the converter cost is 40 units, for example, the maximum number of converters among all the nodes in \( PW P N J \) approach is only about 33% and 8% of that in \( WP N J \) and \( VWP N J \) approaches, respectively, while the maximum number of converters among all the nodes in \( PW P J \) approach is only about 6% of that in \( WP P J \) approach, as shown in Fig. 11(b). The results have been submitted for publication to IEEE/SPIE BroadNets, 2004 (previously known as OptiComm).

4. Impact of Working Path Routing Algorithms on Cost-Effectiveness of \( p \)-Cycle based Protection for WDM Optical Networks

To dynamically provision services in future survivable DOE network (assume for example \( p \)-cycles are used for protection), we need to study the impact of routing algorithms on the cost-effectiveness of the entire network.

In \( p \)-cycle based protection for optical networks, especially in the case of non-joint optimal working
capacity routing and $p$-cycle configuration studied by various researchers, where working capacity of a traffic demand is routed before $p$-cycles are optimally configured, the routing of working capacity of a traffic demand has invariably been assumed to use the shortest path routing. However, shortest paths for routing working capacities may not necessarily result in the best overall network resource optimization when $p$-cycle configuration is figured in. In this work, we conduct simulation based studies of the impact of accommodating working capacities using 7 different routing algorithms on the overall network resource optimization in $p$-cycle protected optical WDM networks. The overall resource optimization problem is solved based on the work reported in [15, 17], where the problems of optimal $p$-cycle configuration in survivable WDM optical mesh networks with no wavelength conversion, full wavelength conversion, and partial wavelength conversion are formulated. Our results show that when wavelength conversion is allowed, the widest shortest path routing, in general, outperforms other algorithms for routing working capacities to achieve the least overall network cost under different optimization models, network configurations, and workloads. The details of simulation results are reported in [10] and submitted for publication to IEEE/SPIE BroadNets, 2004 (previously known as OptiComm). The findings provide a preferred routing algorithm to achieve cost-effectiveness in future survivable networks should $p$-cycle based mechanism is used for network protection.

**Future Challenges**

We have identified a few future challenges:

- **Refining demand model:** An expressive, flexible, and accurate model that captures the application resource demand characteristics is crucial for dynamic resource scheduling and service provisioning. We have an initial model which is to be refined over time with inputs from the application communities as well as the UltraNet researchers. We also plan to adapt the model by using a moving time window to handle dynamically arrived demands, i.e., demands are not given by a known set $M$; rather, a demand dynamically arrives with a requested lasting time.

- **Terabit traffic grooming:** Recognizing that resource demands vary greatly over different applications, e.g., multi-resolution on-demand bandwidths, a group of channels (e.g., a control channel and a data channel) with different on-demand bandwidths, the PI will tackle this problem by designing traffic grooming algorithms in the proposed space-time demand scheduling framework. Our initial work on multicast traffic grooming is reported in [1]. Various provisioning modes must be supported so that applications
requiring multiple channels with a combination of requirements can be effectively supported. The capabilities of provisioned channels must accommodate burst, real-time streams, as well as lower priority traffic.

- **Burst and long session scheduling:** Since the speed of configuration determines routing and connection efficiency, long configuration times are especially not efficient for short burst data transfer. Therefore, until configuration times can be significantly reduced as the technologies advance, an intelligent scheduling should be designed to *hide* the relatively long configuration time. We propose to adapt our proposed space-time scheduling algorithms and techniques currently being developed in the optical burst switching arena to the scheduling of burst data transfers and dynamic on-demand bandwidths.

- **Multicast demand scheduling:** The provisioned channels must also provide for multi-point or shared use to enable increasingly popular distributed large-scale collaboration. A receiver initiated approach is likely to be the most suitable solution. A closely relevant problem is the space-time co-scheduling of unicast channels and multicast channels, and efficient traffic grooming of multi-resolution resource demands. We have done some preliminary work on multicast traffic grooming in [1] as described in the accomplishment section.

- **Delay/jitter agile service provisioning:** Circuit-switched dedicated channels offers the minimum delay and delay jitter. Additional non-deterministic delay and delay jitter may be introduced by host processing (e.g., operating system and protocol processing), network interface cards, access network, and O-E-O conversion. The continental US round trip time is about 40ms. To provide delay/jitter agile services to applications, the PI believes that an adaptive cross-layer approach is needed. Dedicated channels/guaranteed bandwidths are needed to ensure minimum transport delay, especially for instrument control and steering applications. Residual delay jitter must be absorbed in the playout buffer at the receiving end. We propose to investigate and develop appropriate cross-layer design approaches which will take three directions in an integrated manner: (i) low-latency source coding and channel coding, (ii) space-time scheduling of on-demand resources, and (iii) adaptive playout buffering and output scheduling.

- **Integrated framework:** To support DOE’s science mission that include from traditional data services to advanced large scale data transport, a provisioning framework needs to integrate provisioning, scheduling of IP services, non-IP (e.g., dedicated channel) services, delay/jitter agile services, and so on. In addition, resources must also be dynamically provisioned for the production network, research network, and experimental network under the integrated framework.

- **Cost-effective grade of protection:** The network should provide cost-effective grade of protection by allowing the selection from a variety of survivability options (with different associated cost) as needed [11, 7, 9, 8]. The protection schemes should be integrated with the service provisioning framework.

Finally, we hope to collaborate with DOE and UltraNet personnel to design and implement the scheduling server, middleware, and interface with the signaling protocol, test and validate the implementation under controlled deployment in the DOE test-bed network.

**RESEARCH INTERACTIONS**

Our interaction with the DOE researchers have been limited so far given that we have been focused on developing algorithms and conducting basic research. The PI made initial contact with UltraNet personnel, in particular, Dr. Nagi Rao and Dr. Bill Wing at ORNL, on April 6 and 26, seeking feedback on current work and collaboration on dynamic service provisioning. Interactions with DOE researchers will increase in the near
future once the algorithms get more mature and are ready to be tested in a test-bed network. The PI also plan to interact with the DOE application communities to seek information on application requirements and patterns of network use. The goal is to come up with a more suitable model for demands and resource use by DOE applications.

The research interaction and collaboration with Dr. Abdul A. S. Awwal of DoE lab has been very productive. Together, we have identified and solved several interesting problems on unicast/multicast traffic grooming and network protection and restoration [1, 2] that can potentially be applicable to relevant to future DOE networks.

The PI has also started collaboration with Prof. Biswanath Mukherjee of University of California at Davis and Prof. G. K. Chang of Georgia Institute of Technology on issues related to multicast support, space-time scheduling, and delay/jitter agile service provisioning. Discussions with Prof. Mukherjee and Prof. Chang have been very thought-provoking. Feedback on the PI’s current work from Prof. Mukherjee and Prof. Chang has been very positive and helpful.

Finally, we attended IEEE INFOCOM 2004 and 7th INFORMS conference, presented some of our research results and findings [19, 11], and exchanged ideas with researchers from other institutions.

References


APPENDIX A
PROJECT PUBLICATIONS


5. Tianjian Li and Bin Wang, “Cost Effective Shared Path Protection for WDM Optical Mesh Networks with Partial Wavelength Conversion.” Accepted for publication in Photonic Network Communications Journal.


Problem 1. Space-Time Demand Scheduling Optimization: Non-blocking Case

The objective is to minimize the network resource use.

(1) The following are given as program inputs:

- \(N\): the set of nodes in the network.
- \(E\): the set of links in the network.
- \(W\): the set of wavelengths on each link.
- \(D=\{r\}\): the set of scheduled traffic demands; \(s_r, d_r\) and \(n_r\) are the source and destination nodes and the number of requested light-paths of demand \(r\), respectively.
- \(\theta_{r=r'}\in\{0, 1\}\): indicates whether demands \(r_p\) and \(r_q\) are overlapping in time (=1) or not (=0) (\(p \neq q\)).
- \(m_{i,j}\in R^+\): the length of link \((i, j)\).
- \(M\): a large number (e.g., maximum demand lasting time).

(2) The problem solves the following variables given a set of connection requests \(D\):

- \(\delta_{r,\lambda}\in\{0, 1, \ldots n_r\}\): the number of light-paths of demand \(r\) using wavelength \(\lambda\).
- \(A_{i,j}^r,\lambda\in\{0, 1\}\): indicates whether the path of demand \(r\) traverses link \((i, j)\) using wavelength \(\lambda\) (=1) or not (=0).
- \(Z_{i,j}^\lambda\in\{0, 1\}\): indicates whether the path of some demands traverse link \((i, j)\) using wavelength \(\lambda\) (=1) or not (=0).

Objective

\[
\text{minimize}\left\{ \sum_{\forall (i, j) \in E} \sum_{\forall \lambda \in W} Z_{i,j}^\lambda \times m_{i,j} \right\}
\]

Subject to \((r \in D, \lambda \in W, (i, j) \in E)\), if not specified otherwise):

Requested capacities of demand \(r\):

\[
\sum_{\forall \lambda \in W} \delta_{r,\lambda} = n_r
\]  

Flow conservation constraints at source nodes:

\[
\sum_{\forall \lambda, (s_r, o) \in E} A_{s_r, o}^r,\lambda = \delta_{r,\lambda}, A_{i, s_r}^r,\lambda = 0, \forall i : (i, s_r) \in E
\]  

Flow conservation constraints at destination nodes:

\[
\sum_{\forall \lambda, (i, d_r) \in E} A_{i, d_r}^r,\lambda = \delta_{r,\lambda}, A_{d_r, o}^r,\lambda = 0, \forall o : (d_r, o) \in E
\]
Flow conservation constraints at other nodes:
\[
\sum_{\forall i (i,j) \in E} A_{i,j}^{r,\lambda} - \sum_{\forall o (j,o) \in E} A_{j,o}^{r,\lambda} = 0, \forall j \in N(j \neq s_\tau, d_\tau) \tag{6}
\]

Wavelength \(\lambda\) on link \((i,j)\) should not be used by two demands if they are overlapping in time:
\[
A_{i,j}^{r,p,\lambda} + A_{i,j}^{r,q,\lambda} + \theta_{p,r,q}^{r,p,\lambda} \leq 2, \forall \tau_{p,r,q} \in D(p \neq q) \tag{7}
\]

Constraints indicating whether wavelength \(\lambda\) on link \((i,j)\) is used by some demands:
\[
Z_{i,j}^{\lambda} \leq \sum_{\forall r \in D} A_{i,j}^{r,\lambda} \tag{8}
\]

\[
|D| \times Z_{i,j}^{\lambda} \geq \sum_{\forall r \in D} A_{i,j}^{r,\lambda} \tag{9}
\]

Subject to \((r \in D, \lambda \in W, (i,j) \in E, \text{if not specified otherwise})\):

Requested capacities of demand \(r\):
\[
\sum_{\forall \lambda \in W} \delta^{r,\lambda} = n_r \tag{10}
\]

Flow conservation constraints at source nodes:
\[
\sum_{\forall o (s_r,o) \in E} A_{s_r,o}^{r,\lambda} = \delta^{r,\lambda}, A_{i,s_r}^{r,\lambda} = 0, \forall i : (i,s_r) \in E \tag{11}
\]

Flow conservation constraints at destination nodes:
\[
\sum_{\forall i : (i,d_\tau) \in E} A_{i,d_\tau}^{r,\lambda} = \delta^{r,\lambda}, A_{d_\tau,o}^{r,\lambda} = 0, \forall o : (d_\tau, o) \in E \tag{12}
\]

Flow conservation constraints at other nodes:
\[
\sum_{\forall i (i,j) \in E} A_{i,j}^{r,\lambda} - \sum_{\forall o (j,o) \in E} A_{j,o}^{r,\lambda} = 0, \forall j \in N(j \neq s_\tau, d_\tau) \tag{13}
\]

Constraints indicating whether demand \(r_p\) and demand \(r_q\) overlap in time:
\[
M \times C_{1}^{p,r,q} \geq \beta_{r,q} - \alpha_{r,p}, \forall \tau_{p,r,q} \in D(p \neq q), C_1 = 1 \text{ if } \beta_{r,q} > \alpha_{r,p} \text{, otherwise} \tag{14}
\]
\[
M \times (1 - C_{1}^{p,r,q}) > \alpha_{r,p} - \beta_{r,q}, \forall \tau_{p,r,q} \in D(p \neq q), C_1 = 1 \text{ if } \beta_{r,q} > \alpha_{r,p} \text{, otherwise} \tag{15}
\]
\[
M \times C_{2}^{p,r,q} \geq \beta_{r,p} - \alpha_{r,q}, \forall \tau_{r_p,r_q} \in D(p \neq q), C_2 = 1 \text{ if } \beta_{r_p} > \alpha_{r,q} \text{, otherwise} \tag{16}
\]
\[
M \times (1 - C_{2}^{p,r,q}) > \alpha_{r,q} - \beta_{r_p}, \forall \tau_{r_p,r_q} \in D(p \neq q), C_2 = 1 \text{ if } \beta_{r_p} > \alpha_{r_q} \text{, otherwise} \tag{17}
\]
\[
\theta_{p,r,q}^{r_p,r_q} \leq C_{1}^{r_p,r_q} \theta_{r_p,r_q}^{r_p,r_q} \leq C_{2}^{r_p,r_q} \theta_{p,r,q}^{r_p,r_q} \geq C_{1}^{r_p,r_q} + C_{2}^{r_p,r_q} - 1, \forall \tau_{r_p,r_q} \in D(p \neq q) \tag{18}
\]

Wavelength \(\lambda\) on link \((i,j)\) should not be used by two demands if they overlap in time:
\[
A_{i,j}^{r,p,\lambda} + A_{i,j}^{r,q,\lambda} + \theta_{p,r,q}^{r,p,\lambda} \leq 2, \forall \tau_{p,r,q} \in D(p \neq q) \tag{19}
\]

Constraints indicating whether wavelength \(\lambda\) on link \((i,j)\) is used by some demands:
\[
Z_{i,j}^{\lambda} \leq \sum_{\forall r \in D} A_{i,j}^{r,\lambda} \tag{20}
\]

\[
|D| \times Z_{i,j}^{\lambda} \geq \sum_{\forall r \in D} A_{i,j}^{r,\lambda} \tag{21}
\]
Problem 2. Space-Time Demand Scheduling Optimization: Blocking Case I

The objective is to minimize the number of demands to be rearranged. If there exist several solutions that result in rearranging the minimum number of demands, the one that uses the minimum network resources, i.e., the minimum number of wavelength-links (or cost), is selected.

(1) The following are given as program inputs:

- $N$: the set of nodes in the network.
- $E$: the set of links in the network.
- $W$: the set of wavelengths on each link.
- $D=\{r\}$: the set of scheduled traffic demands; $D_r = (s_r, d_r, n_r, \alpha_r, \beta_r, p_r)$ is a tuple representing demand $r$ where $s_r$, $d_r$, $n_r$, $\alpha_r$, $\beta_r$ and $p_r$ are the source node, the destination node, the number of requested light-paths, the setup and tear-down dates, and the priority of demand $r$, respectively. When demand $r$ has a high priority ($p_r=1$), it must be scheduled between $\alpha_r$ and $\beta_r$, i.e., it cannot be rearranged. Otherwise, demand $r$ may be rearranged if its priority is low ($p_r=0$).
- $m_{i,j} \in R^+$: the length of link $(i,j)$.
- $M$: a large number.

(2) The problem solves the following variables given a set of connection requests $D$:

- $\delta^r, \lambda \in \{0, 1, ..., n_r\}$: the number of light-paths of demand $r$ using wavelength $\lambda$.
- $A^r_{i,j} \in \{0, 1\}$: indicates whether the path of demand $r$ traverses link $(i,j)$ using wavelength $\lambda (=1)$ or not ($=0$).
- $Z^r_{i,j} \in \{0, 1\}$: indicates whether the path of some demands traverse link $(i,j)$ using wavelength $\lambda (=1)$ or not ($=0$).
- $\eta^r_\alpha \in \{0, 1, 2,...\}$: the time difference between actual schedule and requested schedule of demand $r$ (for either setup or tear-down date) if demand $r$ is rearranged ahead of schedule.
- $\eta^r_\alpha \in \{0, 1, 2,...\}$: the time difference between actual schedule and requested schedule of demand $r$ (for either setup or tear-down date) if demand $r$ is postponed.
- $\varepsilon^r_\alpha \in \{0, 1\}$: indicates whether demand $r$ is rearranged ahead of schedule ($=1$) or not ($=0$).
- $\varepsilon^r_\alpha \in \{0, 1\}$: indicates whether demand $r$ is postponed ($=1$) or not ($=0$).
- $C^r_{i, p, q} \in \{0, 1\}$: indicates whether the actual tear-down date of demand $r_q (=\beta_{r_q}^+ + \eta^q_{r_q} - \eta^r_{r_q})$ is later than the actual setup date of demand $r_p (=\beta_{r_p} + \eta^p_{r_p} - \eta^r_{r_p})$ $(=1)$ or not $(=0)$ ($p \neq q$).
- $C^r_{p, q} \in \{0, 1\}$: indicates whether the actual tear-down date of demand $r_p (=\beta_{r_p} + \eta^p_{r_p} - \eta^r_{r_p})$ is later than the actual setup date of demand $r_q (=\beta_{r_q} + \eta^q_{r_q} - \eta^r_{r_q})$ $(=1)$ or not $(=0)$ ($p \neq q$).
- $\theta^r_{p, q} \in \{0, 1\}$: indicates whether demands $r_p$ and $r_q$ are overlapping in time $(=1)$ or not $(=0)$ ($p \neq q$).
Objective

\[
\text{minimize}\left\{ M \times \sum_{r \in D} (\varepsilon_r^- + \varepsilon_r^+) + \sum_{(i,j) \in E} \sum_{\lambda \in W} Z_{i,j}^\lambda \times m_{i,j} \right\}
\]  
(22)

Subject to \( (r \in D, \lambda \in W, (i,j) \in E, \) if not specified otherwise):

Requested capacities of demand \( r \):

\[
\sum_{\lambda \in W} \delta_r^\lambda = n_r
\]  
(23)

Flow-conservation constraints at source nodes:

\[
\sum_{w \in (s_r, o) \in E} A_{s_r,o}^r = \delta_r^\lambda, A_{s_r,o}^r = 0, \forall i : (i, s_r) \in E
\]  
(24)

Flow-conservation constraints at destination nodes:

\[
\sum_{w \in (i,d_r) \in E} A_{i,d_r}^r = \delta_r^\lambda, A_{i,d_r}^r = 0, \forall o : (d_r, o) \in E
\]  
(25)

Flow-conservation constraints at other nodes:

\[
\sum_{w \in (i,j) \in E} A_{i,j}^\lambda - \sum_{w \in (j,o) \in E} A_{j,o}^\lambda = 0, \forall j \in N(j \neq s_r, d_r)
\]  
(26)

Constraints indicating whether the actual tear-down date of demand \( r_q \) is later than the actual setup date of demand \( r_p \):

\[
M \times C_1^{r_p,r_q} \geq (\beta_{r_q} + \eta_{r_q}^+ - \eta_{r_q}^-) - (\alpha_{r_p} + \eta_{r_p}^+ - \eta_{r_p}^-),
\]
\[
M \times (C_1^{r_p,r_q} - 1) < (\beta_{r_q} + \eta_{r_q}^+ - \eta_{r_q}^-) - (\alpha_{r_p} + \eta_{r_p}^+ - \eta_{r_p}^-), \forall r_p, r_q \in D(p \neq q)
\]  
(27)

Constraints indicating whether the actual tear-down date of demand \( r_p \) is later than the actual setup date of demand \( r_q \):

\[
M \times C_2^{r_p,r_q} \geq (\beta_{r_p} + \eta_{r_p}^+ - \eta_{r_p}^-) - (\alpha_{r_q} + \eta_{r_q}^+ - \eta_{r_q}^-),
\]
\[
M \times (C_2^{r_p,r_q} - 1) < (\beta_{r_p} + \eta_{r_p}^+ - \eta_{r_p}^-) - (\alpha_{r_q} + \eta_{r_q}^+ - \eta_{r_q}^-), \forall r_p, r_q \in D(p \neq q)
\]  
(28)

Constraints indicating whether demands \( r_p \) and \( r_q \) are overlapping in time (both \( C_1^{r_p,r_q} \) and \( C_2^{r_p,r_q} \) are 1):

\[
\theta^{r_p,r_q} \leq C_1^{r_p,r_q}, \theta^{r_p,r_q} \leq C_2^{r_p,r_q}, \theta^{r_p,r_q} \geq C_1^{r_p,r_q} + C_2^{r_p,r_q} - 1, \forall r_p, r_q \in D(p \neq q)
\]  
(29)

Wavelength \( \lambda \) on link \((i,j)\) should not be used by two demands if they are overlapping in time:

\[
A_{i,j}^{r_p,\lambda} + A_{i,j}^{r_q,\lambda} + \theta^{r_p,r_q} \leq 2, \forall r_p, r_q \in D(p \neq q)
\]  
(30)

Constraints indicating whether wavelength \( \lambda \) on link \((i,j)\) is used by some demands:

\[
Z_{i,j}^\lambda \leq \sum_{r \in D} A_{i,j}^r, |D| \times Z_{i,j}^\lambda \geq \sum_{r \in D} A_{i,j}^r
\]  
(31)

Constraints indicating whether demand \( r \) is rearranged ahead of schedule:

\[
M \times \varepsilon_r^- \geq \eta_r^-, 1 - \varepsilon_r^- > -\eta_r^-
\]  
(32)

Constraints indicating whether demand \( r \) is postponed:

\[
M \times \varepsilon_r^+ \geq \eta_r^+, 1 - \varepsilon_r^+ > -\eta_r^+
\]  
(33)

Demand \( r \) can only be rearranged (advanced or postponed) when its priority is low \((p_r = 0)\):

\[
\varepsilon_r^- + \varepsilon_r^+ + p_r \leq 1,
\]  
(34)
Problem 3. Space-Time Demand Scheduling Optimization: Blocking Case II

The objective is to minimize the total schedule length (i.e., from the instant when the first demand is scheduled to the instant when the last demand finishes). If there exist several solutions which result in the minimum schedule length, the one that uses the minimum network resources, i.e., the minimum number of wavelength-links (or cost), is selected. However, significant number of low priority demands may potentially be rearranged in order to minimize the total schedule length in this problem formulation.

1) Additional variables:

- \( \alpha_{\text{min}} \): the earliest setup date among all the demands.
- \( \beta_{\text{max}} \): the latest tear-down date among all the demands.

Objective

\[
\text{minimize } \{ M \times (\beta_{\text{max}} - \alpha_{\text{min}}) + \sum_{\forall (i,j) \in E} \sum_{\forall \lambda \in W} Z_{i,j}^\lambda \times m_{i,j} \} \tag{35}
\]

2) Additional constraints:

\[
\alpha_{\text{min}} \leq \alpha_r + \eta_r^+ - \eta_r^- , \tag{36}
\]

\[
\beta_{\text{max}} \geq \beta_r + \eta_r^+ - \eta_r^- . \tag{37}
\]