Distributed lightpath establishment techniques using Multi-wavelength Reservation Protocols in WDM optical networks

Malabika Sengupta a,*, Swapan Kumar Mondal a, Debashis Saha b

a Department of Computer Science and Engineering, Kalyani Government Engineering College, Kalyani, West Bengal, India
b IIM, Calcutta, India

Abstract

Distributed lightpath establishment in WDM optical networks involves, apart from routing, two important steps, namely wavelength selection and wavelength reservation. In uni-wavelength reservation protocols, often multiple connection requests unknowingly compete for the same wavelength, even when other free wavelengths are available, resulting in a collision. Attempting multiple wavelengths for reservation in that case improves the probability of successful reservation as exhibited in Destination Initiated Multi-Wavelength Reservation Protocol (DIMRP). So we extend the concept of multi-wavelength reservation to Split Reservation Protocol (SRP) next and then to Markov-based Split Reservation Protocol (MSRP) and find that performance is improved considerably for them too. Initially, we discuss three multi-wavelength schemes separately and compare performance of these protocols with related uni-wavelength protocols in their own category. Then we undertake a comparative study among the three proposed multi-wavelength protocols to find that multi-wavelength MSRP performs the best.

1. Introduction

In distributed WDM optical networks [1–8] having no wavelength conversion facility [9], usually a dedicated lightpath is first established between the source–destination pair, before the actual data transfer starts. A continuous path, having the same wavelength reserved in all the hops of the path, is called a lightpath. Currently, to deal with the growth of optical networks and to make room for a dynamic allocation of lightpaths, distributed schemes have been proposed and are being standardized in the framework of Generalized Multi-Protocol Label Switching (GMPLS) [10]. In GMPLS, the physical wavelengths on each link are represented by generalized labels, and lightpaths are established using the resource reservation protocol with traffic engineering extensions (RSVP-TE) [11–12].

In this work, we have used the network without any wavelength converter. So, first of all a lightpath is to be established between source and destination, and then, that may be used to transfer data. An example is given below to explain lightpath.

Here a network with 5 nodes is taken, as shown in Fig. 1. In this example, for a request Q1 (say) node 1 is source and node 5 is destination. A route R (say) between nodes 1 and 5 is: {(1–3), (3–4), (4–5)}. Lightpath establishment means hop by hop reservation of any one wavelength, throughout the route, between source and destination. In this example, a wavelength \( w_3 \) is reserved for all the hops of the route. So a lightpath in the route R (shown in figure with dotted line) using \( w_3 \) is established.

Lightpath establishment in a distributed system involves three basic steps [1]: (i) routing [16], (ii) wavelength selection and (iii) wavelength reservation. The RSVP-TE protocol is responsible for provisioning the requested lightpaths. Here, we have not dealt with routing, rather concentrated on steps (ii) and (iii). To exemplify our cases, we have considered fixed shortest path routing. However, the distributed lightpath establishment protocols discussed here in terms of (ii) and (iii) are independent of step (i) i.e., they may use other routing methods also.

Lightpath is first established and then used for transmission. After completion of data transfer, the lightpath is torn by releasing the wavelength used in the lightpath.

During lightpath establishment, requests may be blocked due to the unavailability of a common wavelength on all the links of a route. Again, unavailability of wavelength may occur due to scarcity of resource or resource contention. This resource contention is caused by the concurrent attempts by two or more requests to reserve the same wavelength on a common link. Initially, to handle this issue, three basic reservation protocols, were suggested: (i) Source Initiated Reservation Protocol (SIRP) [1] – also called Forward Reservation Protocol (FRP) [2], (ii) Destination Initiated Reservation Protocol (DIRP) [13] – also known as Backward Reservation Protocol (BRP) [2], and (iii) Intermediate Node Initiated Reservation Protocol (INIRP) [14]. It is also reported [8] that...
method uses path to be reserved concurrently, is denoted by tempted for reservation. The number of wavelengths attempted for wavelength reservation is the number of wavelengths attempt of wavelength(s) throughout the route. An important aspect of wavelength(s) reservation is the number of wavelengths reserved prior to actual data transfer. If reservation is initiated from node 1, i.e., from destination. For INIRP, some intermediate node, i.e., node 3 or node 4 may initiate reservation of wavelength.

Basically resource contention occurs due to lack of updated information regarding availability of wavelength. So, in distributed networks, periodical information flooding or exchange of information among neighboring nodes is done to reduce resource contention. However, due to propagation delay, information about wavelength availability is difficult to be guaranteed at any particular place and time in a large distributed system such as a WDM network. So we cannot totally exclude wavelength collision. Collision of wavelength occurs when two or more requests try to reserve the same wavelength. In Fig. 1, if there be a request Q2 (say), having a route R2 (say) between node 2 (source) and node 5 (destination) as: ((2–4), (4–5)), and if Q2 selects the same wavelength (as that of Q1), i.e., w3, then the two requests Q1 and Q2 will face wavelength collision in their common link (4–5).

To handle the resource contention, wavelength selection process can too play a very important role for distribution of wavelengths (a critical resource) and hence blocking of requests. The basic target of wavelength selection process is to select disjoint sets of wavelengths by concurrent requests for a common link.

Though different methods are used for selection of wavelength for reservation, two familiar methods are: random-fit and first-fit [9]. In random-fit, a wavelength is selected randomly from the available pool of wavelengths. In first-fit, all wavelengths are indexed in an order, and the wavelength having the lowest index is selected from the available set of wavelengths for reservation. In another method, wavelength selection is done using label prioritization [15, 21], where the priorities of wavelengths are set depending on their duration of stay in the pool. Accordingly, Giorgetti et al. proposed two schemes [10]:- suggested label (SL) and suggested vector (SV). Both schemes keep track of potential contents using an additional bit in the control messages, sent in the forward direction, for contention detection. The SV scheme additionally keeps record of the previously suggested labels. Thus, the destination node is provided with an indication of potential contents, which helps identification of the wavelength to be preferably reserved. It is reported that the proposed label preference schemes are effective in reducing the blocking probability (denoted by ‘bp’ in this paper) in backward path, without affecting the bp in forward path, under a variety of network loads. Another important selection method uses Markov model first showcased as Markov based Backward Reservation Protocol (MBRP) in [17]. MBRP is discussed in Section 2.

After selection of wavelength, attempts are made for reservation of wavelength(s) throughout the route. An important aspect for wavelength reservation is the number of wavelengths attempted for reservation. The number of wavelengths attempted to be reserved concurrently, is denoted by aggressiveness (denoted by ‘b’ in this paper) [2]. Usually, the reservation protocols [5, 14, 15–19] are least aggressive, i.e., they attempt only one wavelength at a time for reservation (i.e., b = 1). We call them uni-wavelength reservation techniques. On the contrary, multi-wavelength reservation schemes, discussed in this work, attempt to reserve multiple wavelengths simultaneously (i.e., b > 1). In Fig. 1, for Q1, if three wavelengths, w3, w5 and w6 (say) are attempted for reservation simultaneously, then that becomes an example of multi-wavelength approach. Multi-wavelength reservation schemes increase the probability of successful reservation of at least one wavelength throughout the route. But the disadvantage is that they always thrive for over reservation. If pre-reservation of wavelength(s) is done for longer duration, i.e. much before the actual use of them, that phenomenon is called over reservation. Thus multi wavelength reservation may be good for the request under consideration, but it may reduce the probability of successful reservation for the other contemporary requests. Thus, b is required to be managed carefully; otherwise the current request may reserve excessive resource (wavelength) for itself thereby causing scarcity of resource for other requests. Hence, the value of b is to be optimized to achieve the effectiveness of multi wavelength reservation.

In order to demonstrate the efficacy of multi-wavelength approach, previously we applied this concept of using optimized b to DIRP, and the scheme was reported as Destination Initiated Multi-wavelength Reservation Protocol (DIMRP) [13]. In DIMRP, b wavelengths (where b > 1), are attempted for reservation (subject to availability), and thereby bp is reduced.

In this work, we have proposed the use of multi-wavelength approach on two other protocols: (i) Split Reservation Protocol (SRP) [18] and (ii) Markov-based Split Reservation Protocol (MSRP) [19]. The proposed protocols are named as SRP with multi-wavelength (SRPM) and Multi-wavelength MSRP (MMSRP) [20] respectively. In SRPM, splitting may take place depending on some network parameters during probing [2]. If splitting takes place, two reservation packets are initiated in both forward and backward directions simultaneously. Among them, forward reservation packet attempts to reserve b wavelengths to implement aggressiveness. If backward reservation packet fails at some intermediate node, it releases the reserved wavelength and retries with another wavelength from the available set. In MMSRP, an ordered set of wavelengths is obtained using probabilistic method and the set is continuously updated during probing in the forward path before first splitting. If the current wavelength being attempted for reservation is found unavailable, then it uses the top candidate from the ordered set for further splitting.

We have simulated all the above three aggressive schemes and compared their performances.

The paper is organized as follows. Section 2 describes the related reservation protocols and their limitations. Variables and control packets used in different schemes are described in Section 3. Multi-wavelength reservation schemes, DIMRP, SRPM and MMSRP are discussed in Section 4. Section 5 deals with comparison of the protocols and finally, Section 6 concludes the work.

2. Related works

In SIRPs [2], reservation of wavelength is initiated from source, much before the wavelength is actually used for data transmission. This increases the reservation duration (the duration for which wavelength is reserved prior to actual data transfer). If reservation duration is more, wavelength is reserved for a longer period of time, which in effect causes over reservation. Due to over reservation other concurrent requests may not get enough free wavelengths, and hence, overall blocking increases.

To reduce over reservation, DIRPs were proposed [2], where reservation is initiated from destination after successful probing from
source to destination (in the forward path). Probing is done to check the availability of wavelength(s). But successful probing does not always guarantee the availability of wavelength during reservation from destination to source (in the backward path). A particular wavelength, which was available during probing, may not be available, while being attempted for reservation. The selected wavelength may be occupied by some other request within the vulnerable period. Vulnerable period is the time difference between probing a link in the forward path and attempt of reservation in that link in the backward path for a request. The uncertainty in availability of wavelength during reservation increases with increase in vulnerable period. To reduce the vulnerable period, wavelength(s) may be reserved much earlier in the forward path, but this again leads to over reservation. Thus, reservation should be done in such a way that both vulnerable period and reservation duration are traded off. This is one of the most important challenges for any reservation scheme.

In SIRP, reservation duration is high, whereas vulnerable period is nil. In DIRP, after completion of probing, reservation is attempted starting from destination, causing larger vulnerable period. To minimize the effect of both (reservation duration and vulnerable period), reservation may be initiated from intermediate node and such protocols are referred as INIRP. If initiation of reservation is done statically from some specified nodes, the scheme is reported as Intermediate node Initiated Reservation Protocol (IIRP) [14]. The specified nodes are predefined in the network through domain based selection method. Also the source and destination are always treated as specified nodes. It is reported that, IIRP performs better than DIRP, which in turn, performs better than SIRP [14]. This IIRP is basically static INIRP, where for all requests, reservation is attempted from pre-defined specified intermediate nodes, and hence it may invite over reservation.

Later dynamic INIRP is proposed and the scheme is reported as SRP. In this scheme, reservation is initiated from intermediate node only when required and the intermediate node is selected dynamically. The concept of splitting shortens the vulnerable period and hence reduces the probability of blocking. It is also reported that SRP performs better than IIRP, as far as bp is concerned.

Selection of wavelength is another important aspect of reservation protocols to reduce wavelength conflict among contemporary requests. To address this aspect, Markov-based Backward Reservation Protocol (MBRP) was proposed [17], where selection of wavelength is done using Markov method and reservation is done like BRP (i.e. DIRP). In this method, one wavelength is selected using Markov model and that is used as guessed_wavelength. This guessed_wavelength remains tagged with a particular request, so that other requests cannot select that wavelength for their use. Thus, wavelength conflict among contemporary requests is reduced. It is reported that MBRP performs better than DIRP. When this Markov method of selection of wavelength is used on SRP, the scheme is reported as MSRP as mentioned earlier. In MSRP, reservation of wavelength is done using dynamic splitting, like SRP. So MSRP performs better than its peers.

However, a major limitation of all the above protocols is that they attempt only one wavelength for reservation. So, only one chance is used by a request. If the selected wavelength fails during reservation, the request is blocked. Hence, there is still scope for further improvement by extending wavelength selection to multiple wavelengths. This scope motivates us to use a set of wavelengths for probable selection and further splitting, if required for failure cases. This concept helps many erstwhile failure cases succeed, thereby reducing the overall blocking of requests. These protocols are reported as multi-wavelength versions of their uni-wavelength siblings with appropriate modifications.

3. Variables and control packets

We define the following terms to be used in the subsequent sections of the paper.

- \( R \): a route consisting of \( n \) number of nodes, \( node_0 \) through \( node_{n-1} \) (\( n > 1 \)), where \( node_0 \) is the source and \( node_{n-1} \) is the destination.
- \( N \): number of wavelengths per fibre.
- \( connection_id \): identity number of a request.
- \( source \): the first node of \( R \), where a request comes.
- \( source_id \): identity number of source of a request for \( R \).
- \( destination \): the last node of a route \( R \).
- \( destination_id \): identity number of destination of a request for \( R \).
- \( intermediate \): any node except the source and destination of \( R \).
- \( present \) node: the node (say node\(_0\)) of a route where the control packet under consideration has reached. It may be noted that any node of a route may become a present node at a given point of time.
- \( next \) node: the node next to the present node (node\(_0\)) i.e., node\(_{i+1}\) if the movement of the control packet is considered towards destination. If the movement of the control packet is considered towards source, then the next node will be node\(_{i-1}\).
- \( previous \) node: the node previous to the present node (node\(_0\)) i.e., node\(_{i-2}\) if the movement of the control packet is considered towards source. When the movement of the control packet is considered towards source, it is node\(_{i-1}\).
- \( present \) link: the link which connects the present node and the next node of a route.
- \( previous \) link: the link which connects the previous node and present node of a route.
- \( pre_hop_id \): identity number of previous node in \( R \) of a request.
- \( next_hop_id \): identity number of next node in \( R \) of a request.
- \( arrival_time \): time when the request arrives at the present node.
- \( guessed_wavelength \): the wavelength guessed in the present link by a request at present node.
- \( link_id \): identity number of a link.
- \( bit_map \): represents status of usage of all wavelengths of the link. A ‘1’ is placed in the bit_map when the corresponding wavelength is free and a ‘0’ otherwise. The size of bit_map equals to the number of wavelengths used in the links.
- \( rate_map \): contains rate of change of states of the wavelength usage for all the wavelengths in the corresponding link_id.
- \( route-path \): the ordered list of nodes on a selected route.
- \( node_table \): keeps records of all requests passing through the node. Each record of node_table contains the following attributes of all the requests passing through the node: connection_id, source_id, destination_id, pre_hop_id, next_hop_id, arrival_time and guessed_wavelength. The duration of a record in a node_table is bounded by the estimated source–destination round trip time of the concerned request.

We assume that each node broadcasts its adjoining link usage information at every \( T \) seconds [17]. This link usage information is stored in node_link_status_table at every node. The records in node_link_status_table contain the attributes: link_id and bit_map.

Markov_table contains the information of rate of change of states of the wavelength usage for all the wavelengths in all the links. The records of markov_table contain the following: link_id and rate_map.

The space complexity of each of the above mentioned tables is \( O(N) \), where \( N \) is the total number of wavelengths per link.

Different control packets used in the protocols discussed here are described below. All the control packets contain following common fields: source_id, destination_id, route_path and connection_id.
So these fields are not mentioned again while packets are described.

PROB: moves from source towards destination. It contains the additional field: prob_set, which is an array indicating the availability/unavailability of each wavelength in the route. MSRP and MMSRP use one more field called prev_guess_index, which stores the guessed_wavelength.

RES: moves towards source to reserve the selected wavelength(s) available as reserve_set in it.

REL: moves towards destination or splitting point (sp) to release the reserved wavelength(s). The node where PROB splits, is called the splitting point.

RES_FWD: moves towards destination to reserve the selected wavelength(s) available as reserve_set in it.

RES_FWD_REL: moves towards destination to reserve the selected wavelength(s) available as reserve_set in it; it also reserves the wavelength(s) available as reserve_set in it. It may contain a field future_guess_set which contains the wavelengths marked for future reservation, if required.

ACK: moves towards source, carries acknowledgement of RES_FWD.

NACK: moves towards source, carries not acknowledgement of RES_FWD or PROB.

NACK_REL: moves towards source to release the reserved wavelength(s), carries not acknowledgement of RES_FWD.

DATA_TRANS: transmits data from source, releases the wavelengths used for transmission, also releases the selected wavelength(s) reserved (if any).

4. Protocols using multi-wavelengths

In this section, we discuss the three protocols which use multi-wavelengths, namely DIMRP (multi-wavelength version of DIRP), SRPM (multi-wavelength version of SRP) and MMSRP (multi-wavelength version of MSRP).

4.1. DIMRP

In DIMRP, when a request arrives, source initiates the usual control packet PROB, which moves towards the destination. If PROB reaches destination successfully, then destination checks the availability of wavelength(s) in prob_set of PROB.

If the number of available wavelengths in prob_set is \( \alpha \) (say), destination randomly selects \( \min(b, \alpha) \), number of wavelengths from prob_set to obtain reserve_set, where \( b \) is a positive integer such that, \( 1 < b < \alpha N \). (\( N \) is the number of wavelengths per link).

The reserve_set contains the set of wavelengths to be attempted for reservation. From destination, RES starts, and moves towards source, carrying reserve_set. RES attempts to reserve the wavelength(s) included in reserve_set, for every link present on its way. If on the way, at any link, a wavelength is not available, the wavelength is dropped from reserve_set. If RES successfully reaches the source with non-empty reserve_set, then it is a case of success. If \( \alpha \) number of wavelengths present in reserve_set is \( p \), (where \( p < b \)) then from \( p \), source selects one wavelength randomly for data transmission. Then RES is converted into DATA_TRANS for transmission of data. The timing diagram, presented in Fig. 2, describes this case of success. During probing, if prob_set becomes empty, then PROB is converted into NACK, which moves towards source. After the NACK reaches the source, the request is blocked. Again, during reservation, if the reserve_set becomes empty due to non-availability of wavelength, at some intermediate node, then RES is converted into REL, which moves towards destination and releases the reserved wavelengths. Also from that node, a NACK is generated which moves towards source. The node where a reservation attempt fails, is denoted as fp. When the NACK reaches the source, the request is blocked. This case of failure, during reservation, is shown in Fig. 3.

It may be noted that multi-wavelength (aggressive) reservation approach must be optimized and to achieve this, \( b \) can be varied from 1 to \( N \). If \( b = 1 \), the probability of successful reservation remains low, on the other hand, if \( b = N \), over reservation may spoil the advantage of multi-wavelength reservation. Thus, finding an optimum value of \( b \), is a challenging issue. The optimum value of \( b \) depends on mean arrival rate of connection requests (cr) and \( N \). Henceforth we will use the word requests to represent connection requests.

We have simulated the protocols considering static shortest path routing in a randomly generated mesh network of 40 nodes and 46 links. Requests arrive following Poisson’s distribution, and connection holding times are exponentially distributed. The simulation model is event driven. The source and destination of a request are chosen at random. We have also used 95% confidence level for statistical validation in all simulation results.

On the basis of simulation results, DIMRP is compared with its uni-wavelength sibling DIRP, also referred as Destination Initiated Single wavelength Reservation Protocol (DISRP). Fig. 4 shows variation of \( \text{bp} \) with \( \text{cr} \) for \( w = 75 \) and \( b = 3 \). The figure shows that DIMRP is a clear outperformer over DISRP. This is because success rate of reservation is improved through multi-wavelength reservation attempt in DIMRP.

4.2. SRPM

In general, for INIRP, vulnerable period is reduced at the cost of reservation duration. In static INIRP, reservation of wavelength is attempted unconditionally from some predefined nodes. The
predefined nodes are obtained through domain based selection [14]. This approach suffers from extreme cases because reservation is initiated unconditionally from pre-selected intermediate nodes. For example, for a particular request, if a predefined node exists next to the source of the route, it initiates backward reservation after traversing one hop only and the case is almost like DIRP. Hence, it suffers from over reservation resulting increase in bp. Similarly, if the first predefined node exists just before destination, the case is like DIRP, when the first RES is initiated in backward path when only one hop is left. Thus, probability of getting any free wavelength is reduced due to long vulnerable period. In dynamic INIRP i.e., SRP, the nodes, from where the initiation of reservation takes place, are not predefined, rather selected dynamically.

To reduce the vulnerable period and hence the bp, concept of splitting is introduced. ‘Splitting’ means splitting of PROB into two RES packets (RES_FWD and RES_BKD) at some intermediate nodes. Thus, reservation can be attempted in both directions (towards source and destination) simultaneously.

However, splitting also invites certain degree of over reservation. Due to splitting, simultaneous reservations in forward path (towards destination, using RES_FWD) and backward path (towards source, using RES_BKD) take place, and thus, vulnerable period and reservation duration for RES_FWD and RES_BKD become different. For RES_FWD, vulnerable period is zero but for RES_BKD, vulnerable period depends on the position of splitting. It may be noted that, if splitting occurs nearer to destination, vulnerable period increases for backward reservation but the reservation duration in forward direction becomes less. On the other hand if splitting occurs near source, vulnerable period for backward reservation decreases but causes more over reservation for forward reservation. Thus, position of splitting is to be optimized in order to reduce the effect of both over reservation and vulnerable period. Hence the node where the splitting takes place may be decided dynamically using appropriate system parameters. This type of splitting is called dynamic splitting and it is used in the protocol, SRPM, explained below.

When a request comes, source initiates a PROB, which moves towards the destination. PROB includes hop_count and b along with other fields. The variable b is a predefined positive integer. Basically b is the aggressiveness of the scheme, which is used for taking decision of splitting. For the first link, prob_set is initialized to the wavelengths available in the first link. For the subsequent links, on receiving PROB, a node performs two tasks:

- updates the prob_set using the operation, prob_set = prob_set \cap \text{available wavelength(s) on the present link}
- checks the conditions for probable splitting.

SRPM dynamically splits probe attempt, into two concurrent (one towards source and the other towards destination) reservation attempts, at any intermediate node. For a request, splitting may occur, if the following two conditions are satisfied:

(a) \( (x_1+d) \leq \text{hop_count} \leq (x_2+d) \) i.e., whether the PROB has traversed more than a pre-selected distance \( (x_1+d) \) of the route as well as less than another preselected distance \( (x_2+d) \), where \( d \) is total number of hops of the route and \( x_1 \) and \( x_2 \) are two positive fractions within 0 and 1.

(b) number of available wavelengths of prob_set \( \leq b \), for \( b > 1 \).

The variation of \( x \) (i.e., \( x_1 \) and \( x_2 \)) is studied in [18]. We select, \( x_1 = 0.5, x_2 = 0.6 \), so that vulnerable period as well as reservation duration is optimized to have low bp.

If conditions of splitting are satisfied, splitting occurs. PROB is converted into two reservation packets: RES_FWD and RES_BKD. At sp, reserve_set is copied into both RES_FWD and RES_BKD. RES_BKD selects one wavelength (say \( w_1 \)) randomly from the reserve_set and moves towards source, and RES_FWD moves towards destination, attempting to reserve all wavelengths of reserve_set. RES_BKD attempts to reserve only one wavelength to reduce the problem of over reservation but RES_FWD reserves all the wavelengths of reserve_set to facilitate the retries if required. However, if the reserve_set is empty, request fails and subsequently the request is blocked.

If RES_BKD successfully reaches the source, then at source, it waits for ACK of RES_FWD. If RES_FWD reaches the destination, with non-empty reserve_set, then an ACK is sent towards source along with the reserve_set. ACK, on its way, keeps a copy of the reserve_set at sp, for future use in case of retries. After receiving the ACK, the source checks the matching of the wavelength reserved in forward and backward directions. If those are matched, the data transmission starts. If there be mismatch in wavelength reservation or if RES_BKD fails, then RES_BKD is converted into REL which moves towards sp releasing the reserved wavelength. At sp, REL randomly selects another wavelength from the reserve_set for retry, and becomes RES_BKD again. This is repeated (if required) until total number of retries \( (=b-1) \) are exhausted.

Figs. 5 and 6 show the case of success without retry and with one retry respectively. If RES_FWD is stuck at some intermediate node before destination, the attempt of reservation fails (Fig. 7). Then RES_FWD is converted into NACK_REL. The NACK_REL moves from that fp to the source, and releases the wavelength reserved by both RES_FWD (from fp to sp) and RES_BKD (from sp to source). After receiving the NACK_REL at source, the request is blocked.

Performance of SRPM is compared with its peer IIRP. In case of IIRP, different wavelengths are attempted for reservation from
different intermediate nodes, which is equivalent to retries. SRPM using two retries, is compared with IIRP, and variation of \( bp \) with \( cr \) is shown in Fig. 8. From the figure, we find that, for both the schemes, \( bp \) increases with \( cr \) due to increase in crisis of wavelength. Also, we see that SRPM outperforms IIRP throughout the range of \( cr \) shown in the figure. Thus, the protocol, SRPM can be considered as better performer than IIRP with respect to \( bp \).

4.3. MMSRP

MMSRP is basically multi-wavelength version of MSRP. MMSRP also uses Markov model for selection of wavelength. Since Markov model uses Markov chain to describe each state of wavelength usage, so maximum allowable transition is one. Thus, multiple numbers of wavelengths cannot be reserved at a time. In MMSRP, a set of wavelengths (instead of one) is selected by Markov model and continuously updated for possible future use. In case of failure, during reservation in the backward direction, it retries to reserve the next best wavelength, through another splitting at the failure point. Thus, MMSRP handles multiple wavelengths sequentially through multiple splitting. The protocol is discussed here. In MMSRP, each node broadcasts two types of information regarding its entire adjoining links: (i) link usage information at every \( T \) seconds, (ii) transition rates of all states at every \( T \) seconds. Though link usage information is broadcast at a regular interval (\( T \)), but that is not necessarily correct at an arbitrary time between \( sT \) and \((s+1)T\). Where \( s \) is a positive integer. To overcome this uncertainty, a prediction is suggested to select wavelength during these intervals. To take the probabilistic method of selection, a C–T Markov chain is used in this work. The states of Markov chain is defined as the number of used wavelengths in a link, i.e. \( ith \) state represents that \( i \) number of wavelengths are used. Thus, each link remains in any one of the total \( N+1 \) number of states (\( N \) is already defined as total number of wavelengths per link). The transition rates of all states, are considered as Markov parameters. These parameters for each link are broadcast at every \( T \) second and stored in a table referred as markov_table at all nodes. So essentially markov_table contains the information of rate of change of states of the wavelength usage for all the wavelengths in all the links. \( T \) is considered to be much longer compared to \( T \). If value of \( T \) is lower than a certain level, it is vulnerable to oscillation which may ultimately lead to poor performance.

Since \( T_{ratio} \) (the ratio of \( T \) to \( T \)) is an important parameter which affects the performance of the protocol, it is studied for different set of values of \( cr \) and \( wl \). It is found that an optimum value of \( T_{ratio} \) exists in each case. It is reported that \( 17,19 \) values of \( T_{ratio} \) corresponding to minimum value of \( bp \) is near 300. Hence, for simulation results, the optimum value of \( T_{ratio} \) is kept as 300.

In this protocol, when a request arrives at a node, the node guesses multiple wavelengths based on the link usage information of the previous link and the markov_table. The wavelengths thus guessed, at that instant of time have the maximum probability of remaining available throughout the route. If total number of free wavelengths is \( y \), a node selects \( b \) (\( y \geq b > 1 \)) number of wavelengths having higher probabilities of availability. These \( b \) number of wavelengths are arranged with respect to probability in descending order as \( \lambda_1, \lambda_2, \ldots, \lambda_b \). Obviously, if \( y < b \), all \( y \) wavelengths are selected. From the ordered set, wavelength \( \lambda_g \) is selected as guessed_wavelength and wavelengths \( \lambda_2 \) to \( \lambda_g \) are stored in future_guess_set.

When the present node is the source, prev_guess_index of PROB is initialized to \( \lambda_g \). A record is created in node_table of the source and PROB is forwarded to next node. If the present node is any node other than source, the node checks the availability of the wavelength stored in prev_guess_index (i.e., \( \lambda_g \)). If the wavelength is available, a record is created in node_table of the present node and PROB is forwarded to next node; else prev_guess_index is updated to the wavelength \( \lambda_g \) and future_guess_set is also updated to include next (\( b-1 \)) number of wavelengths. Then the node checks the condition for splitting. When splitting does not occur, a record is created in node_table of the present node and the PROB is forwarded to next node.

Thus, each node after receiving the PROB, performs the following major tasks for the request: (i) detects the wavelengths already guessed by earlier requests and excludes them from prob_set, (ii) guesses multiple wavelengths for this request from the remaining free wavelengths and updates PROB, (iii) initiates on-demand splitting (dynamically) if necessary.
MMSRP adaptively splits a probe attempt into two concurrent (upstream and downstream) reservation attempts at some intermediate node selected dynamically. For a request, if hop_count is the number of hops traversed by the PROB, then, splitting may occur provided both the following conditions are satisfied:

(i) \((x_1 + d) \leq \text{hop}\_\text{count} \leq (x_2 + d)\) i.e., whether the PROB has traversed more than a pre-selected distance \((x_1 + d)\) of the route as well as less than another preselected distance \((x_2 + d)\), where \(d\) is total number of hops of the route, \(x_1\) and \(x_2\) are two positive fractions within 0 and 1, and \(x_2 > x_1\).

(ii) The wavelength at \text{prev\_guess\\index} is different from \(\lambda_{sp}\).

If the conditions of splitting are satisfied, splitting occurs; otherwise the PROB propagates to the nest node. As mentioned earlier the values of \(x_1\) and \(x_2\) are selected as 0.5 and 0.6 respectively.

If splitting occurs, the PROB is converted to RES_FWD. A RES_BKD is also generated at the first splitting point (we call it \(sp_1\)) as shown in Figs. 9 and 10. RES_BKD includes the fields: connection_id, selected_wavelength and future_guess_set. At the point of splitting, \text{prev\_guess\\index} of PROB is assigned to selected_wavelength of both RES_FWD and RES_BKD. The RES_BKD moves towards the source, reserving the wavelength stored at \text{selected\\wavelength}, \(\lambda_{g1}\), and deletes the entries of this request in node\_tables on the way. The RES_FWD moves towards destination reserving the same wavelength \(\lambda_{g1}\).

However, if RES_BKD fails at some intermediate node due to non-availability of \(\lambda_{g1}\) further splitting may occur (maximum \(b\)-1 times). In that case, the node selects next candidate from the future_guess_set, i.e., \(\lambda_{g2}\), subject to availability both in previous link and present link. Then RES_BKD again splits into two new reservation packets RES_FWD and RES_BKD. These RES_FWD and RES_BKD, act like earlier RES_FWD and RES_BKD packets respectively. Both packets attempt to reserve the selected wavelength in the same way, in both forward and backward directions. RES_FWD on its way also releases the previously reserved wavelength by previous RES_FWD and RES_BKD.

If both RES_FWD and RES_BKD are successful to reserve the same wavelength, data transmission starts (Fig. 9) after receiving the ACK from \textit{destination}. If RES_BKD is stuck at an intermediate node and all possible splittings are exhausted, the request is blocked and RES_BKD is converted into NACK which moves towards source. Another REL is generated from the point of failure which moves towards \textit{destination} and releases the wavelength reserved so far by both RES_FWD and RES_BKD. Again, if RES_FWD fails, it is converted into REL_BKD which moves towards source releasing the wavelengths reserved so far. It also acts as a NACK and deletes the entries of this request in node_tables of the nodes it passes through. Fig. 9 shows a case of success whereas Fig. 10 shows a case of failure. In the figures, \(sp_1\) and \(sp_2\) indicate the two splitting points (nodes) of a request.

We have studied MMSRP exhaustively and compared with its peers. Some representative results are presented here. Fig. 11 represents the variation of \(\text{bp}\) with \(\text{cr}\) for \(wl = 500\). In general, as expected, for all the schemes, \(\text{bp}\) increases with increase in \(\text{cr}\).

From the figure, we find that MMSRP always performs better than MSRP and MBRP. Also we can observe that relative performance is much better for higher values of \(\text{cr}\). It happens because, at high load, the crisis of getting a wavelength is more, and in case of MMSRP a reasonable number of failure cases may succeed using future_guess_set and multiple splitting, leading to improvement in \(\text{bp}\).

Fig. 12 represents variation of \(\text{average latency}\) with \(\text{cr}\), for \(wl = 500\). \(\text{Average latency}\) is considered only for successful requests. It can be observed from the figure that \(\text{average latency}\) is decreasing with the increase in \(\text{cr}\) for all the protocols. This happens because, as \(\text{cr}\) increases, success rate for long haul routes reduces, and more successful short haul requests contribute to the average latency. However, success rate for long haul requests in MMSRP is more compared to other two schemes yielding relatively higher average latency.

Fig. 13 shows the variation of \(\text{average control overhead}\) with \(\text{cr}\), for \(wl = 500\). Due to inherent characteristics of splitting in dynamic INIRP schemes (MSRP and MMSRP), more control packets are used in these two schemes compared to MBRP. Again MMSRP uses more control packets compared to MSRP to tackle multi-wavelength guessing.
5. Comparison of protocols

In the previous sections, we have discussed different multi-wavelength protocols, where multiple wavelengths are used for lightpath establishment. The protocols are separately compared with their peers and their performances are found to be quite promising. It may be noted that, while multiple wavelengths are selected for reservation, the multiplicity is always kept within a threshold value, beyond which multi-wavelength reservation based protocols may not perform in a better way. In this section, we have presented comparative study of these three multi-wavelength based protocols, DIMRP, SRPM and MMSRP.

Since the key performance parameter in lightpath establishment is $bp$, we have studied $bp$ exhaustively. Relative performance of MMSRP for $wl = 200$ is shown in Fig. 14. From the figure, it is found that, in general, for all the protocols, $bp$ increases with increase in $cr$. MMSRP performs distinctly better than other two protocols beyond $cr = 40$. Also it can be observed from Fig. 14 that, with the increment of $cr$, the relative performance of MMSRP also improves. This happens because, as $cr$ increases, crisis also increases and selection of wavelength becomes more important to avoid collision of wavelength. This situation is tackled in a better way through wavelength guessing in MMSRP. Another important thing is that MMSRP uses the future_guess_set, through the process of multiple splitting, and this successfully reduces $bp$.

However, for low value of $cr$ and $wl$, DIMRP performs better than other two schemes (Fig. 15) because at this region advantage of aggressiveness in DIMRP becomes more effective. This happens due to the fact that, at low demand of wavelengths, the effect of over reservation due to aggressiveness becomes insignificant, compared to the advantage of multiplicity.

6. Conclusion

In uni-wavelength distributed reservation protocols, often multiple requests unknowingly compete for the same wavelength, even when other free wavelengths are available, resulting in a collision. Attempting multiple wavelengths in that case improves the probability of successful reservation. However, this concept invites over reservation because, too much network resource may be used (through pre-reservation of multiple wavelengths) by one request. In such cases, future requests may be blocked due to non-availability of wavelengths. In this paper, we have addressed those issues in the multi-wavelength version of three important reservation protocols, namely DIRP, SRP and MSRP. The new protocols are, termed as DIMRP, SRPM and MMSRP, respectively. These protocols are described and compared with their corresponding single wavelength versions. Finally, the performances of multi-wavelength protocols are compared.

Since MMSRP uses multiple splitting, combined with multi-wavelength guessing, it reduces $bp$ considerably, compared to both SRPM and DIMRP. During probing, first splitting is used dynamically, and in case of failures, further multiple splittings are used.
during reservation. Thus, MMSRP performs better than its peers at higher wavelength regions as far as blocking probability is concerned. So it may be considered as a better performer in WDM networks, specially for the applications, where protocol efficiency is of prime importance and the network uses a larger number of wavelengths (e.g., dense WDM networks).

Acknowledgement

The first author is supported by the project, Development of Efficient Protocols for Distributive WDM Optical Networks, sponsored by AICTE, Govt. of India.

References