A Scalable Wavelength Assignment Algorithm Using Minimal Number of Wavelength Converters in Resilient WDM Networks

Technical Report UTD/EE/4/2009

September 2009

Miguel Razo, Shreejith Billenahalli, Wanjun Huang, Arularasi Sivasankaran,
Limin Tang, Hars Vardhan, Marco Tacca, and Andrea Fumagalli
OpNeAR Lab, Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas, Richardson, TX, USA
Paolo Monti
NeGONet Group, School of Information and Communication Technology, ICT-FMI
The Royal Institute of Technology, Kista, Sweden

Abstract

Careful wavelength assignment (WA) to support lambda services is necessary to reduce the total number of wavelength converters (WCs), which are required every time the wavelength continuity constraint cannot be met in wavelength division multiplexing (WDM) networks. With the successful introduction of reconfigurable optical add-drop multiplexers (ROADMs) and related technologies, WDM networks are now growing in size, both in the number of optical nodes and number of wavelengths supported, thus requiring WA algorithms that scale with the network size.

This paper presents a scalable and efficient WA algorithm that aims to reduce the total number of WCs in WDM networks bearing static lambda services. The WA algorithm is applicable to both unprotected and (dedicated) protected lambda services. In the latter case, wavelength continuity constraint between the working and the protection path is taken into account. The WA algorithm is then used to quantify the tradeoff between using tunable optical transceivers versus number of WCs to cope with the wavelength continuity constraint.

Keywords: Wavelength Converters, Wavelength Assignment, WDM networks

1. Introduction

End-to-end optical circuits can be established in optical transport networks to avoid electronic processing at the intermediate nodes. In WDM networks, an optical circuit is obtained by reserving a wavelength channel from the source to the destination node. If possible, wavelength continuity is preferred when reserving the wavelength channel for the circuit, across all the fibers traversed by the circuit path. When the wavelength continuity constraint cannot be satisfied due to the already reserved wavelengths, a wavelength converter (WC) is necessary along the optical circuit, to shift the signal from the available wavelength in one fiber to the available wavelength in the next fiber. Due to their relatively high cost, WCs should be used minimally.

The problem of assigning wavelengths (WA) to optical circuits in some optimal way has been long studied. [5] showed that the min-RWA (optimal way of routing and wavelength assignment) is NP-hard. In [3], it was shown that the WA problem is equivalent to the coloring problem, under the assumption that WCs cannot be used. [8] also shows the applicability of partition coloring for wavelength assignment. In [2], [1], solutions were proposed to compute both routing and wavelength assignment (RWA) for unprotected lambda services, in order to minimize a given cost function. This cost function can be defined to minimize the number of necessary WCs for each lambda service, given the set of unreserved wavelengths. Quite a number of additional papers have addressed the problem of minimizing the blocking probability of dynamically created optical circuits, for a given set of existing WCs in the network [4]. A newly generated optical circuit is blocked when neither the wavelength continuity constraint can be met nor WCs are available in the network to circumvent the wavelength continuity constraint. More recently, a number of solutions have been proposed to generalize both WA and RWA problems to account for optical transmission impairments [7, 9, 10].

With the successful introduction of reconfigurable optical add-drop multiplexers (ROADMs) and related technologies, WDM networks are now growing in size, both in the number of optical nodes and wavelengths supported, thus requiring WA algorithms to scale with the network size. When facing the problem of minimizing the number of WCs required to support a number of static lambda services, the available existing solutions [2], [6], [11] do not scale well due to the size of the graph (that is required in the computation), which is a function of the product between the number of nodes and number of wavelengths. Another limitation of these solutions is their applicability to unprotected lambda services only.

This paper presents a scalable and efficient WA algorithm that aims to reduce the total number of WCs in WDM networks bearing static lambda services. The WA algorithm is applicable to both unprotected and (dedicated) protected lambda services. In the latter case, wavelength continuity constraint between the working and the protection path is taken into account. The WA algorithm extends the coloring solution in [3] to take into account the possibility of using WCs. Given a precomputed routing for the lambda services, the WA algorithm runs in two steps. In the first step, the algorithm assigns one wavelength to every lambda service (both unprotected and protected) that can meet the wavelength continuity constraint. A simple graph coloring algorithm is used in this phase, leaving the lambda services that cannot meet the wavelength continuity constraint uncolored. In the second step, every uncolored lambda service is assigned two or more wavelengths along with the necessary WCs to enable wavelength hopping. The wavelengths are assigned by their popularity (the wavelength available in maximum number of fibers in most popular) in this step. In the results section a simple routing technique is used for the purpose of testing the WA algorithm for scalability. However, the WA algorithm is independent of the end-to-end routing computation and can be used in conjunction with any suitable routing algorithm.

Due to its low complexity, the proposed WA algorithm is shown to be applicable to networks with up to 1500 nodes, and the number of wavelengths per fiber up to 160. On the same computer, the algorithm in [2] cannot work for networks larger than 65 nodes and 40 wavelengths. The optimality of the solution found by the proposed WA algorithm is comparable to the solution found by [2] in a number of experiments conducted by the authors. One application of the proposed WA algorithm is also illustrated in the paper, which leverages its ability to solve the WA problem for protected lambda services.

2. The WA Algorithm

Let the optical transport network be modeled as an undirected graph $G(N, E)$, where N is the set of network nodes, and E is the set of edges. Each edge represents a pair of fibers, one for each direction of propagation. Let each fiber carry up to W wavelengths. Let the set of lambda services contain both a number of bidirectional unprotected services and a number of bidirectional (dedicated path) protected services. Assume that every protected service requires two disjoint paths (link, node or SRLG[1]), i.e., a working path and a protection path. Assume that protection switching is achieved in the optical layer, i.e., the signal is generated by the transmitter (laser) and optically switched to either the working or the protection path (1:1 protection)[2]. The reverse procedure is used at the receiver, to collect the received signal from either the working or the protection path. Assume that the transmitter and receiver are both non tunable. Then the wavelength assigned to both the working and the protection path must be the same, i.e., wavelength continuity constraint. Assume that the routing for both the unprotected and the protected services are given. Bidirectional WCs may be required when a service (unprotected or protected) cannot be assigned the same identical wavelength value across every edge that belongs to the service path(s).

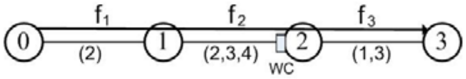

Figure 1

Wavelength conversion along a path.

Fig. 1, depicts wavelength conversion along a path. There is a request from node 0 to node 3 and the request takes the route 0-1-2-3. The unreserved wavelengths (colors) on

---

[1] SRLG is an acronym for Shared Risk Link Groups

[2] For simplicity the proposed algorithm is described for 1:1 dedicated path protection switching, but it also works in the case of 1+1 dedicated path protection switching.

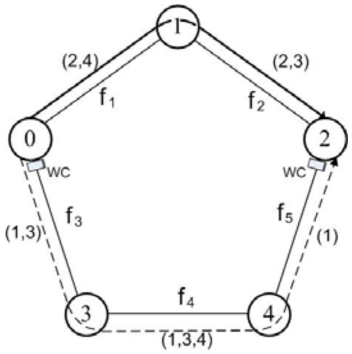

Figure 2

Wavelength conversion at source/destination of a dedicated protected service.

each link of the service request are shown in brackets, next to the link. The wavelength continuity constraint cannot be met for this service. Assume that the first two links (0-1 and 1-2) will be assigned the color 2 and the last link (2-3) will be assigned the color 1. Here a WC is needed at node 2, since there is a change in the wavelength along the path.

Fig. 2, shows wavelength conversion at source/destination of a dedicated protected service. There is a service request from node 0 to node 2. The working path (solid arrow) takes route 0-1-2 and the protection path (dashed arrow) takes route 0-3-4-2. The unreserved wavelengths (colors) on each link of the request are shown in brackets, next to the link. The wavelength continuity constraint cannot be met for this service too. Assume that the links of the working path (0-1 and 1-2) will be assigned the color 2 and the links of the protection path (0-3, 3-4 and 4-2) will be assigned the color 1. If the transceivers are not tunable, they can work only with a single wavelength. Hence wavelength converters are needed at source node 0 and destination node 2.

The WA problem is formulated as follows: Assign each lambda service one or more dedicated wavelength values along its path(s), in order to minimize the total number of WCs, which are required in the network. To solve the WA problem, the following scalable and efficient WA algorithm is used. The algorithm has two steps.

Step 1. In the first step, wavelengths are assigned to all the services, which can be assigned a single wavelength value. This is done by coloring a conflict graph with a technique similar to the one proposed in [3]. Every lambda service is associated with a node in the conflict graph. An edge is added between two nodes of the conflict graph if the paths of the two corresponding services share at least one edge in G. Note that both working and protection paths must be accounted for when the service is protected. Once the conflict graph is built, the coloring is computed using any graph coloring algorithm. For example, the nodes are ordered in non-increasing order of their nodal degree in the conflict graph. Each node is examined in this order and is assigned a color which is not already assigned to any of its neighbors. If no such color is available (only W colors are available, one color corresponding to one wavelength value), the node is left uncolored. The color of the node is then turned into the corresponding wavelength, which is assigned to the service corresponding to the node. The colored services meet the wavelength continuity constraint, and do not require any WCs. Nodes (services) left uncolored require some WCs and are dealt with in step 2. A pseudo-code description of step 1 is given in Algorithm 1.

Step 2. For every service left uncolored after running step 1, the following algorithm is used to assign two or more wavelength values to the service. Consider one uncolored service at a time. Create a subgraph of G, which is formed by only the edges and the nodes

Algorithm 1 Wavelength Assignment Using Graph Coloring

```
Build the graph with service requests as nodes
Add edges between nodes if the service request share a fiber
Sort the nodes in the non increasing order of nodal degree
Initialize all the nodes to be uncolored
for each node in the graph in sorted order do
    for each color from startColor to maxColors do
        if none of the neighbors colored till now has this color then
            Assign this color to the node
            Break the loop
        end if
    end for
end for
```

that belong to the uncolored service path(s). Sort the edges by increasing number of unreserved wavelengths. Sort the wavelength values by decreasing wavelength popularity, i.e., the popularity of a wavelength is defined as the number of edges in the subgraph of G, which have that wavelength value unreserved. Every edge in the subgraph is uncolored and it is then colored using one of its unreserved wavelengths as follows. Start with the uncolored edge, which has the smallest number of unreserved wavelengths. Color that edge with the most popular wavelength value that is available on that edge. Let this wavelength value be the temporary default wavelength. This colored edge forms a fragment of the subgraph of G. The fragment is augmented by adding only edges which can be colored using the default wavelength, as follows. Every edge which is adjacent to the fragment is added to the fragment if the default wavelength is unreserved on the edge. Every edge added to the fragment is colored with the default wavelength. The algorithm continues to add edges to the fragment till no additional edges can be added. The following iterative step is then performed until all edges in the subgraph of G are colored. Create a new fragment with the uncolored edge, which has the smallest number of unreserved wavelengths. Color that edge with the most popular wavelength value that is available on that edge. This wavelength value is the new default wavelength, and this fragment is augmented as described earlier, till no more edges can be added. A pseudo-code description of step 2 is given in Algorithm 2.

The execution of step 2 is illustrated with the help of Fig. 3, which shows the subgraph of G obtained for a protection lambda service. The protection service has source node 0 and destination node 2. The working path (solid arrow) takes route 0-1-2 and the protection

Algorithm 2 Coloring lambda services using multiple colors

```
for each node in the graph that is not colored in Step 1 do
    Rank colors based on the availability in the fibers of the node (The color available in most fibers is top ranked, the one which is least available is ranked lowest)
    Sort the fibers of the node which need to be colored in increasing order of available colors
    for each fiber in the sorted order which has not assigned a color do
        Get the top ranked available color in the fiber
        Assign this color to the fiber
        adjacentFibers := All the fibers adjacent to this fiber
        while adjacentFibers is not empty do
            if the color is available in the fiber then
                Assign this color to the fiber
                Add all the fibers adjacent to this fiber to adjacentFibers, if they have not assigned a color yet
            else
                Remove the fiber from adjacentFibers
            end if
        end while
    end for
end for
```

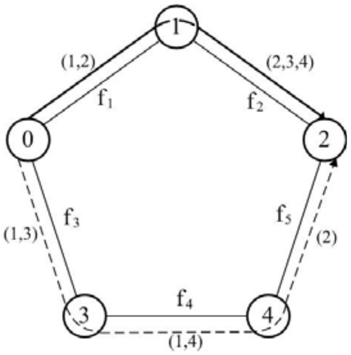

Figure 3

Step 2: Example using a dedicated protected lambda service.

path (dashed arrow) takes route 0-3-4-2. The unreserved wavelengths (colors) on each edge of the subgraph are shown in brackets, next to the edge. The wavelength continuity constraint cannot be met for this service, and step 2 is therefore required. f_1, f_2, f_3, f_4 and f_5 represent the fibers in the corresponding edges. The algorithm first sorts the wavelengths based on their decreasing popularity. Here wavelength 1 is available in 3 fibers, 2 is available in 3 fibers, and so on. The wavelengths are then sorted as 1, 2, 3 and 4. The edges (fibers) are sorted with increasing number of unreserved wavelengths: f_1 has 2 unreserved wavelengths, f_2 has 3, and so on. Edges are thus sorted as follows: f_5, f_1, f_3, f_4 and f_2. f_5 is considered first and is assigned the only unreserved wavelength, i.e., 2. f_5 constitutes the first fragment. All edges which are adjacent to the fragment (f_2 and f_4) are considered next. f_2 can be added to the fragment as it can be assigned wavelength 2 too. f_4 cannot be added to the fragment, as wavelength 2 is not unreserved on that edge. The fragment now contains both f_5 and f_2, and f_1 is an adjacent edge to the fragment. f_1

is added to the fragment and assigned wavelength 2. At this point the fragment cannot be further augmented. The algorithm then creates a new fragment by choosing the uncolored edge with the smallest number of unreserved wavelengths. In this case, both f_3 and f_4 have each 2 unreserved wavelengths. Let f_3 be the chosen edge. f_3 is assigned wavelength 1, which is more popular than wavelength 3. The new fragment is augmented by adding f_4, which is assigned wavelength 1 too. All the edges in the subgraph are now colored and the algorithm stops. In summary, this lambda service is assigned two wavelength values: wavelength 2 over edges f_1, f_2 and f_5 and wavelength 1 over edges f_3 and f_4. Two bidirectional WCs are then required: one at node 4, along the protection path; the other at node 0[3].

3. Numerical Results and Analysis

To investigate the performance of the proposed WA algorithm, a number of experiments are carried out using an Intel P4, 3.4 GHz machine with 1 GB memory. Numerical results are computed by averaging the results of multiple experiments.

The number of wavelengths per fiber is chosen to be $W = 8, 16, 40, 80$ and 160. The fiber layout of the network is randomly generated with the number of nodes ranging from $N = 10$ to 1500. The average nodal degree varies from 2.4 to 9.2. The number of lambda services is chosen to obtain various traffic loads. The source-destination pair for each lambda service is chosen using a uniform distribution over all possible pairs.

---

[3] The transmitter at node 0 can be assigned wavelength 2 (1), and the WC is placed between the transmitter and the multiplexer of f_3 (f_1).

The following solutions are compared. Solution (CFZ-RWA) is obtained running the RWA algorithm in [2]. Solution (CFZ-R+WA) is obtained using the routing solution found by solution (CFZ-RWA), and solving the WA problem using the proposed WA algorithm. Solution (R+WA) is obtained using the simple routing solution described next, and solving the WA problem using the proposed WA algorithm.

Simple Routing Algorithm. A simple routing solution is used to compute the route for each lambda service, which is independent of the routing solution used in [2]. Thanks to its simplicity, the simple routing solution is able to quickly compute routes even in large size networks. The computed routes are then used to test the scalability of the proposed WA algorithm in large networks.

Algorithm 3 describes the simple routing technique. The set of lambda services R and the topology description T are given as input. In this technique the cost of the link is increased each time a new lambda service is routed. If the maximum capacity of the link is reached (i.e., the number of lambda services using the link is equal to the number of wavelengths available on the fiber), then the link is removed from the topology.

As already mentioned, this simple routing technique is just one possible choice, and other efficient routing algorithms can be used for this step.

3.1 Optimality

A network with $N = 30$ nodes and $W = 40$ wavelengths per fiber is considered. The number of bidirectional links (L) is either 71 (Fig. 4) or 133 (Fig. 5). Both figures show the required number of WCs as a function of the number of unprotected lambda services.

Algorithm 3 simpleRouting(R, T)

```
for all lambda services $r \in R$ do
    Compute shortest path from r.source to r.destination in T
    if path is found then
        for all links $l \in path$ do
            if $l.cost <$ number of wavelengths - 1 then
                Increase $l.cost$ by one
            else
                Remove link from T
            end if
        end for
    end if
end for
```

Some conclusions can be drawn. For small nodal degree values (Fig. 4), both solutions (CFZ-R+WA) and (R+WA) are consistently better than solution (CFZ-RWA). For large nodal degree values (Fig. 5), solution (CFZ-RWA) requires fewer wavelength converters compared to both solutions (CFZ-R+WA) and (R+WA) at low and medium loads. At high loads, however, regardless of the nodal degree, solutions (CFZ-R+WA) and (R+WA) do considerably better. Overall, the optimality gain offered by solutions (CFZ-R+WA) and (R+WA) over solution (CFZ-RWA) grows as the number of lambda services increases.

Fig. 6 shows the wavelength converters used by the three approaches described above for $W = 80$ wavelengths. The network under consideration has $N = 20$ nodes and $L = 26$ links. Again with a small nodal degree, solutions (CFZ-R+WA) and (R+WA) require fewer wavelength converters than (CFZ-RWA).

3.2 Scalability

Results shown in Table 1 are obtained by progressively increasing the network size, in terms of number of nodes, links, and lambda services (Req.) with $W = 40$ wavelengths. Solution (CFZ-RWA) cannot be run successfully when the number of nodes is 65 or greater, due to the large size graph required to run the algorithm in [2]. For the same reason, solution (CFZ-R+WA) cannot be found for large networks as the routing for the lambda services cannot be obtained from solution (CFZ-RWA). Solution (R+WA) is able to work with up to 1500 nodes.

The last column in Table 1 shows the time (in seconds) required for wavelength assignment in (R+WA) solution. It can be observed that the algortihm runs in reasonable time, even for large networks.

3.3 Application

A useful application of the proposed algorithm is to quantify the tradeoff between using tunable transceivers versus WCs in provisioning protected bidirectional lambda services. Assume that every lambda service in the network requires one working path and one protection path, which are link disjoint. Assume that a percentage of these lambda services will be provisioned using a pair of tunable transceivers, one at each end node. With the pair of tunable transceivers, it is possible to relax the wavelength continuity constraint across the working and protection path, as each path can be assigned an independent wavelength. This relaxation of the wavelength continuity constraint has a visible impact on the number of WCs, which are required in the network, as quantified in Fig. 7. The figure plots the number of WCs as a function of the number of protected lambda services in a $N = 30$ node and $L = 103$ link network. Five curves are shown, assuming that the percentage of tunable transceiver pairs is, from top to bottom, 0, 25, 50, 75, and 100. The trend reveals that a considerable percentage of WCs can be saved by using a fraction of tunable transceivers. For example, an average of 50.6% WCs can be saved when using 25% tunable transceivers. An average of 77.3% WCs can be saved when using 50% tunable transceivers.

Fig. 8 shows the wavelength converters for $W = 16$ wavelengths and a network with $N = 20$ nodes and $L = 87$ links. Again, the wavelength converters required are considered when 0, 25, 50, 75 and 100% of tunable transceivers are used, and the same behavior can be observed as before.

The last set of results are obtained using $W = 160$ wavelengths and four network topologies, namely case A160,B160,C160 and D160 respectively. Case A160 consists of a network with $N = 10$ nodes and $L = 13$ links, the results are shown in Fig. 9, Case B160 consists of a network with $N = 20$ nodes and $L = 48$ links, the results are shown in Fig. 10, Case C160 consists of a network with $N = 30$ nodes and $L = 38$ links, the results are shown in Fig. 11, Case D160 consists of a network with $N = 30$ nodes and $L = 71$ links, the results are shown in Fig. 12,

4. Conclusions

This paper presented a scalable and efficient WA algorithm that aims to reduce the total number of WCs in WDM networks. The WA algorithm is applicable to both unprotected and (dedicated) protected lambda services. Due to its low complexity, the proposed WA algorithm is shown to be applicable to networks with up to 1500 nodes, and a number of wavelengths per fiber up to 160. On the same computer, comparable algorithms available in the literature cannot work for networks larger than 65 nodes and 40 wavelengths.

Acknowledgments

The authors would like to acknowledge the help received from Poornima Hanumara in testing many of the PlaNet-PTN features.

References

[1] D. Banerjee and B. Mukherjee, "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, 1996, pp. 903–908.

[2] I. Chlamtac, A. Farago, and T. Zhang, "Lightpath (wavelength) routing in large WDM networks," *Selected Areas in Communications, IEEE Journal on*, vol. 14, no. 5, Jun 1996, pp. 909–913.

[3] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath Communications: An Approach to High Bandwidth Optical WANs," *IEEE Transaction on Communications*, vol. 40, no. 7, July 1992, pp. 1171–1182.

[4] X. Chu, B. Li, and Z. Zhang, "A dynamic rwa algorithm in a wavelength-routed all-optical network with wavelength converters," 2003, pp. 1795–1804.

[5] T. Erlebach, K. Jansen, and C. Elvezia, "The complexity of path coloring and call scheduling," *Theoretical Computer Science*, vol. 255, 2001, p. 2001.

[6] S. Gowda and K. M. Sivalingam, "Protection Mechanisms for Optical WDM Networks based on Wavelength Converter Multiplexing and Backup Path Relocation Techniques," *IEEE INFOCOM*, 2003.

[7] Y. Huang, J. P. Heritage, and B. Mukherjee, "Connection provisioning with transmission impairment consideration in optical WDM networks with high-speed channels," *Journal of lightwave technology*, vol. 23, no. 3, 2005, p. 982.

[8] G. Li and R. Simha, "The partition coloring problem and its application to wavelength routing and assignment," 2000.

[9] G. Markidis, S. Sygletos, A. Tzanakaki, , and I. Tomkos, "Impairment aware based routing and wavelength assignment in transparent long haul networks," *Lecture notes in computer science*, vol. 4534 LNCS, 2007, p. 48.

[10] R. Martinez, F. Cugini, N. Andriolli, L. Valcarenghi, P. Castoldi, L. Wosinska, J. Comellas, and G. Junyent, "Challenges and requirements for introducing impairment-awareness into the management and control planes of ASON/GMPLS WDM networks," *IEEE communications magazine*, vol. 44, no. 12, 2006, p. 76.

[11] Y. Wang, L. Li, and S. Wang, "A new algorithm of design protection for wavelength-routed networks and efficient wavelength converter placement," *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 6, 2001, pp. 1807–1811 vol.6.

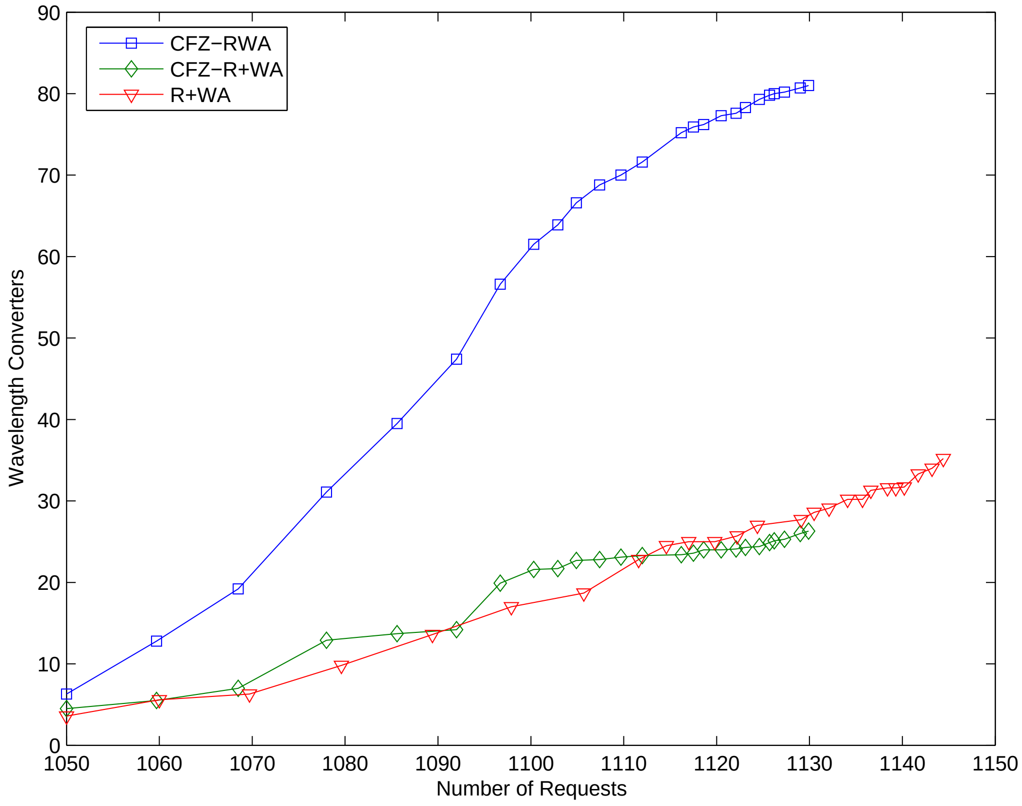

Figure 4

Number of WCs as a function of the total number of unprotected lambda services.

$N = 30$, $L = 71$, and $W = 40$.

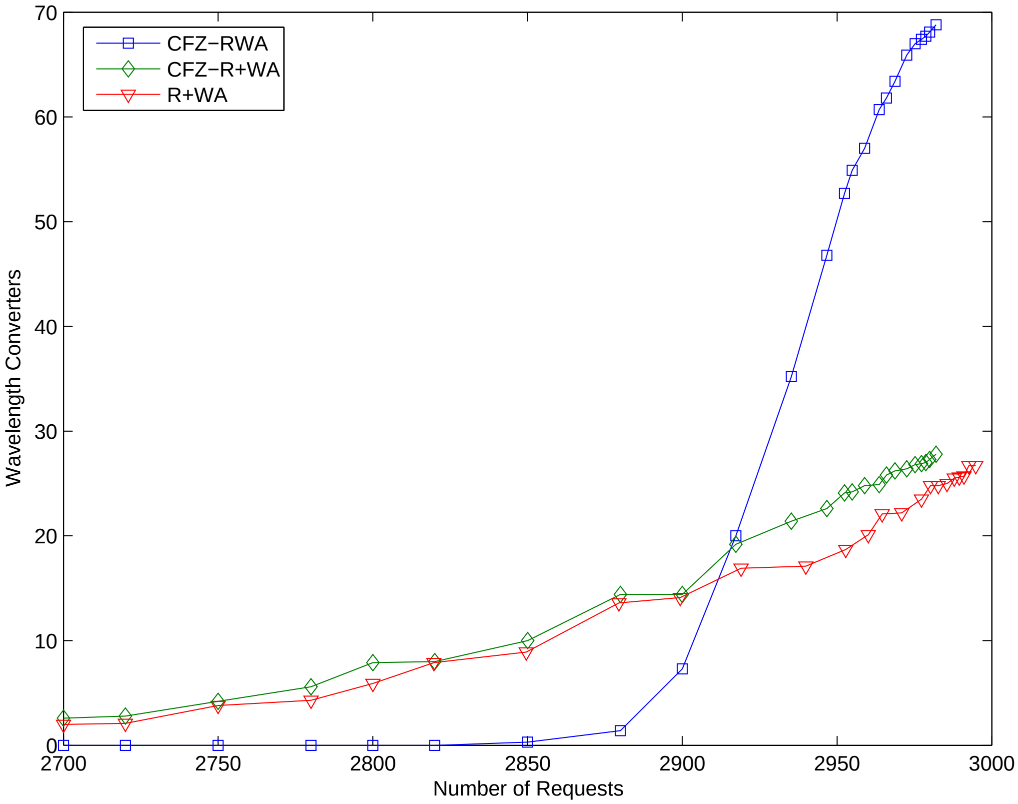

Figure 5

Number of WCs as a function of the total number of unprotected lambda services.

$N = 30$, $L = 133$, and $W = 40$.

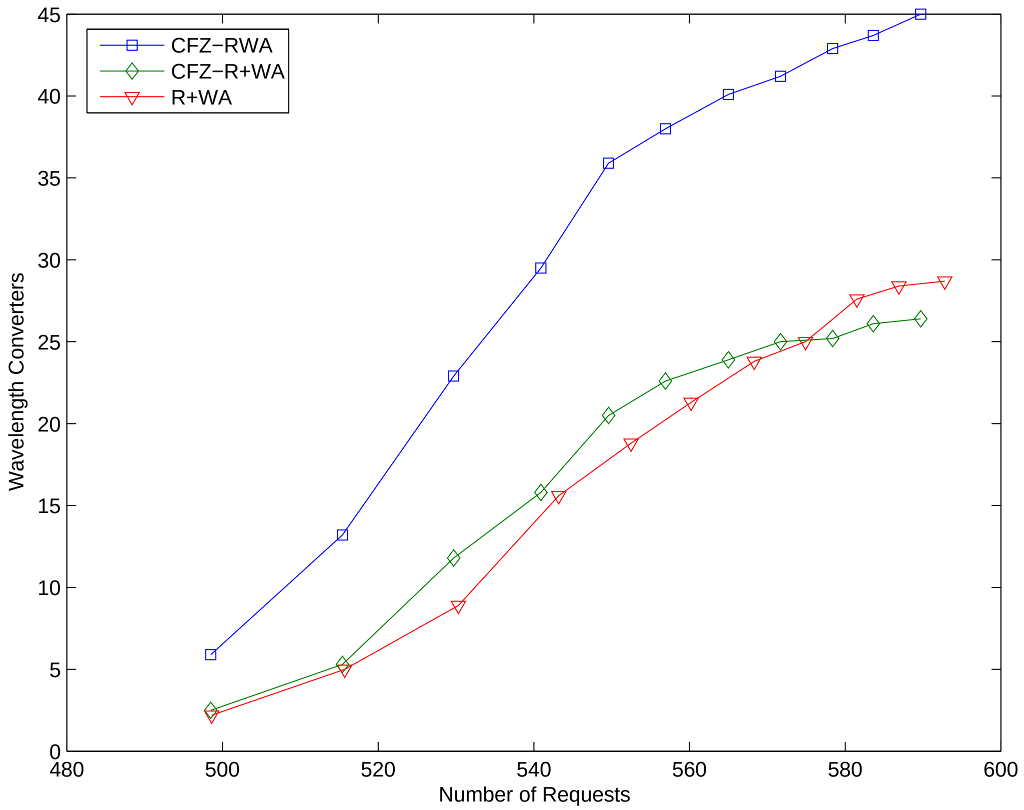

Figure 6

Number of WCs as a function of the total number of unprotected lambda services.

$N = 20$, $L = 26$, and $W = 80$.

Table 1

Results changing network size and traffic load

		CFZ-RWA		CFZ-R+WA	R+WA		
N	L	Req.	WCs	WCs	Req.	WCs	Time[s]
30	133	2966	61	24	2980	25	5
40	179	3675	91	52	3682	51	7
50	223	4296	171	101	4335	116	10
55	246	4606	179.5	112	4646	112	12
60	268	4926	245	155	4988	165	15
75	335	-	-	-	5934	255	19
100	446	-	-	-	7351	413	30
150	666	-	-	-	10142	832	53
200	890	-	-	-	12909	1343	61
300	1337	-	-	-	18063	2564	108
400	1789	-	-	-	23060	3932	174
500	2231	-	-	-	24857	4781	254
600	2676	-	-	-	32261	6860	351
700	3123	-	-	-	36675	8281	470
800	3577	-	-	-	41301	10032	594
900	4029	-	-	-	45816	11659	725
1000	4464	-	-	-	49972	13277	895
1250	5572	-	-	-	60313	17543	1273
1500	6705	-	-	-	69998	18177	1750

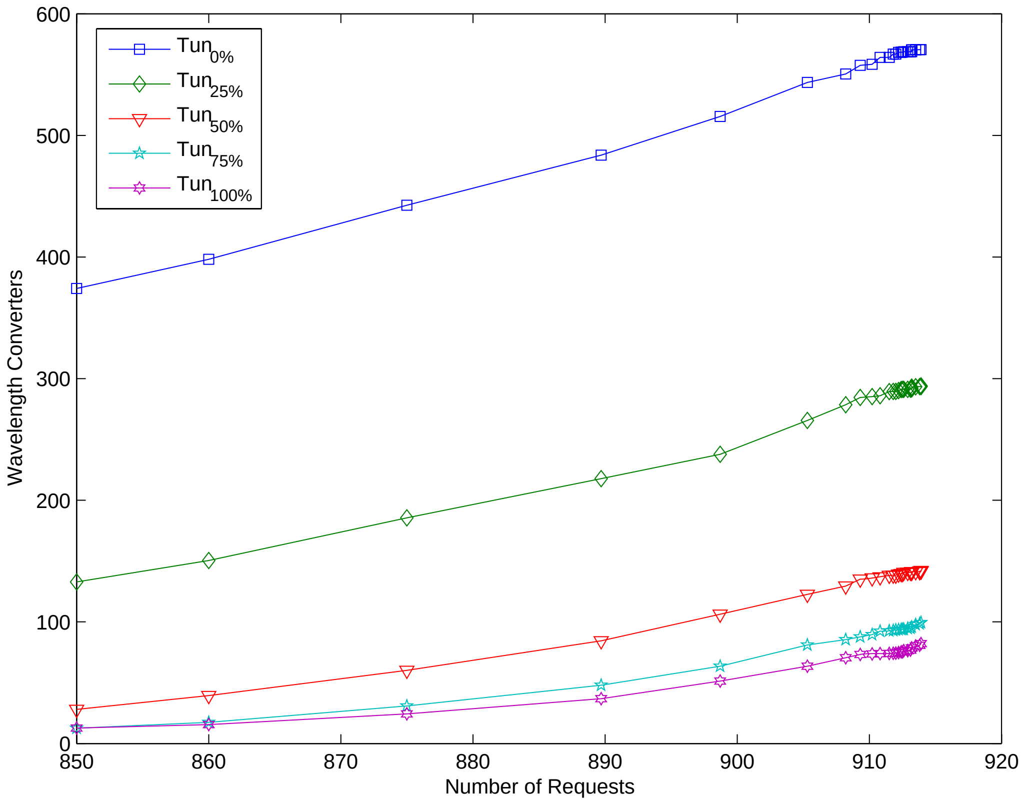

Figure 7

Number of WCs as a function of the total number of dedicated protected lambda services.

$N = 30$, $L = 103$, and $W = 40$.

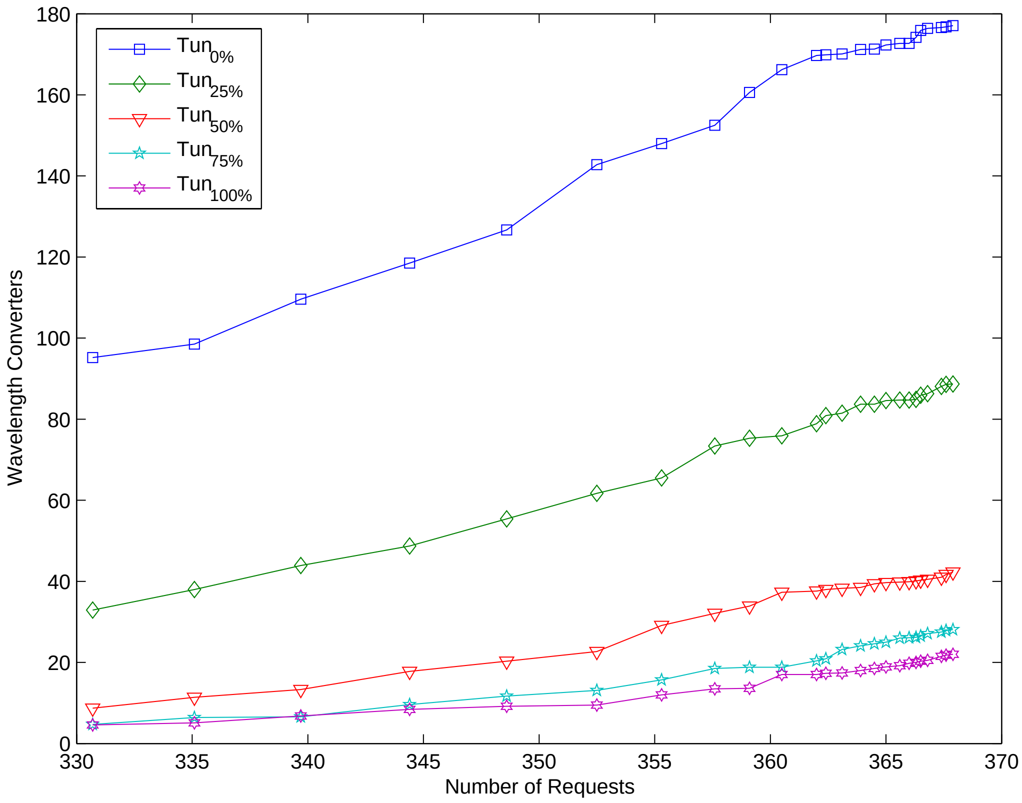

Figure 8

Number of WCs as a function of the total number of dedicated protected lambda services.

$N = 20$, $L = 87.$, and $W = 16$.

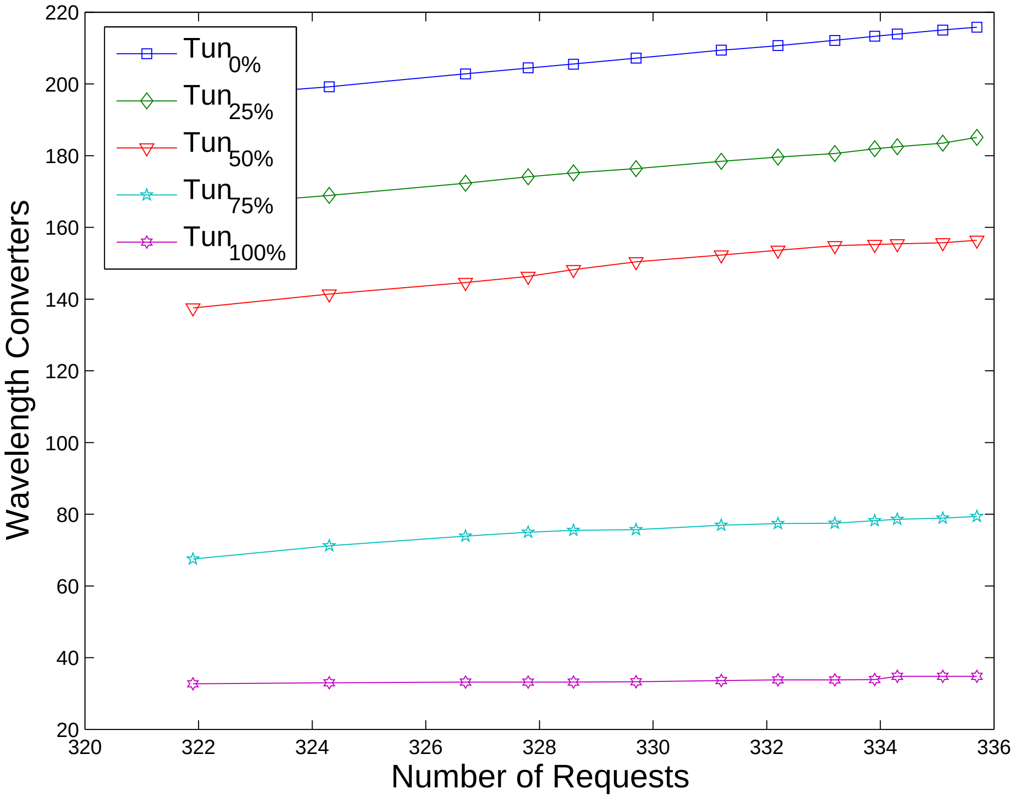

Figure 9

Number of WCs as a function of the total number of dedicated protected lambda services.

$N = 10$, $L = 13.$, and $W = 160$.

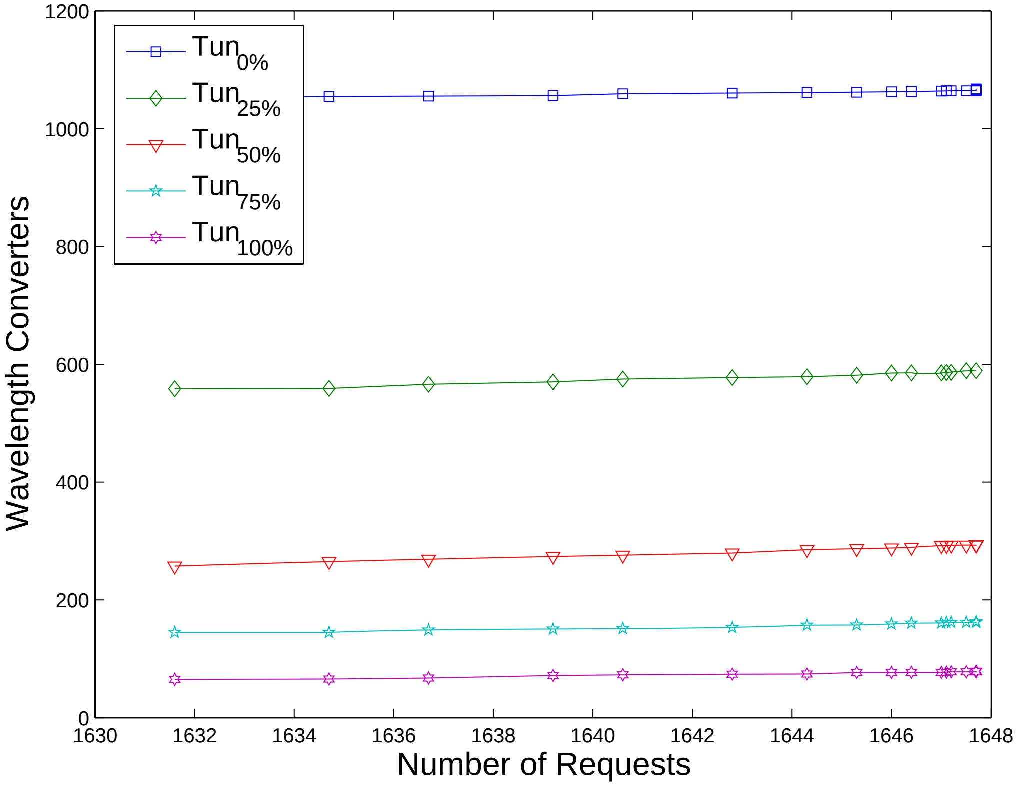

Figure 10

Number of WCs as a function of the total number of dedicated protected lambda services.

$N = 20$, $L = 48$., and $W = 160$.

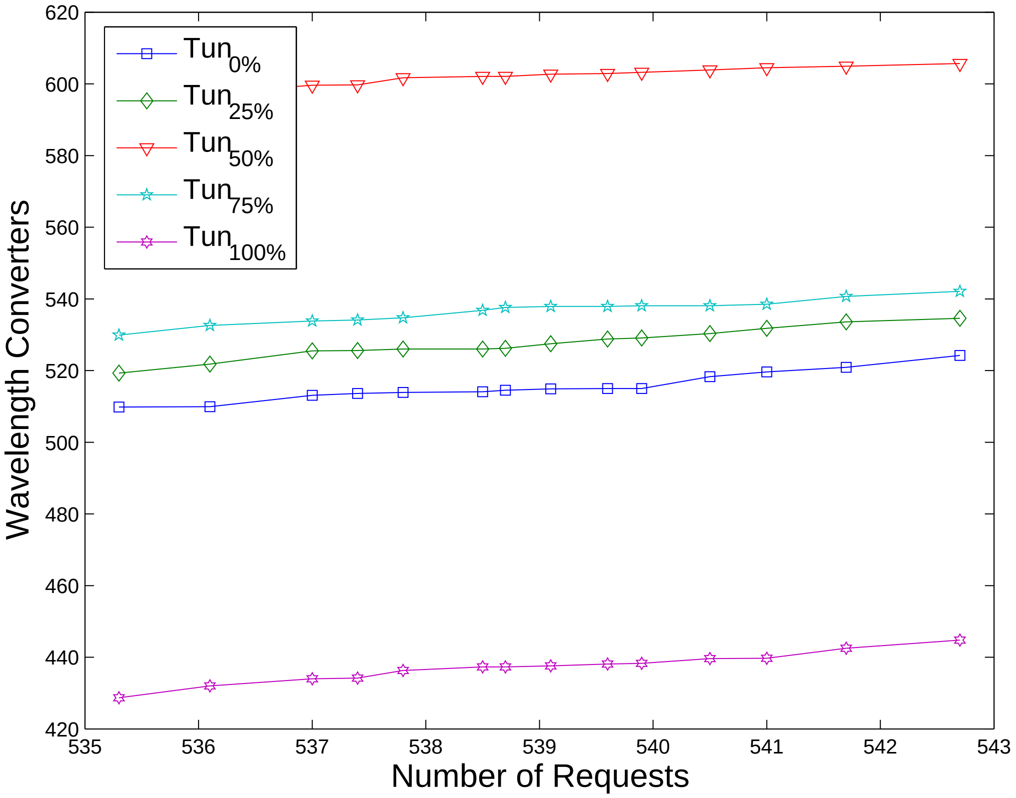

Figure 11

Number of WCs as a function of the total number of dedicated protected lambda services.

$N = 30$, $L = 38.$, and $W = 160$.

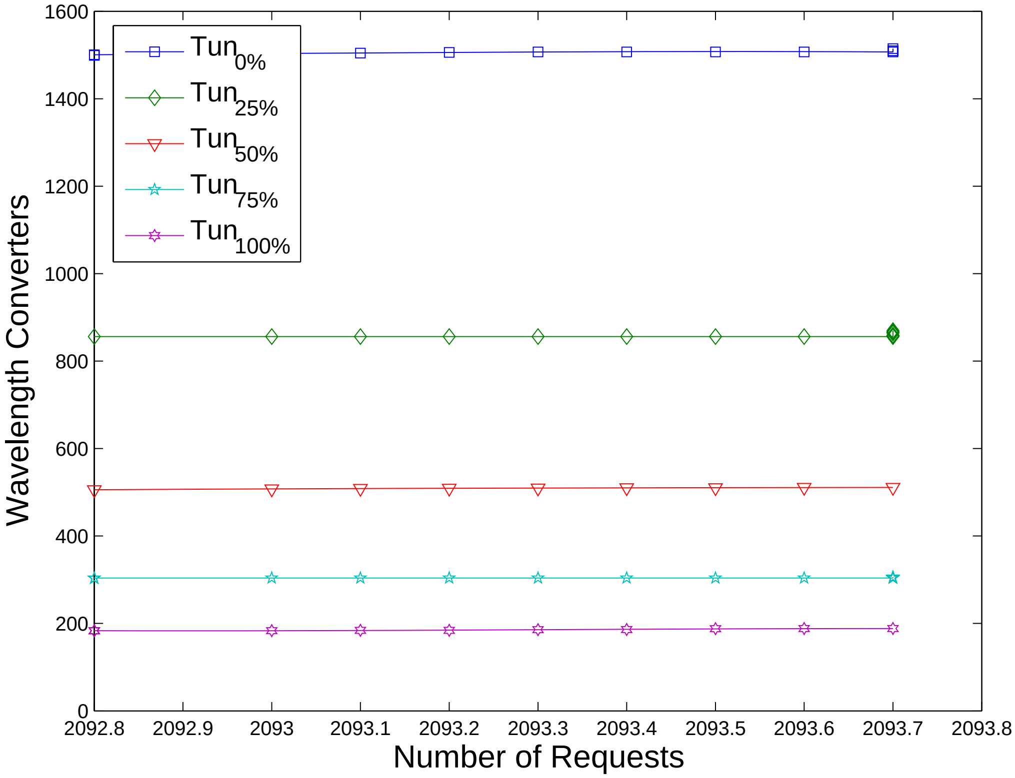

Figure 12

Number of WCs as a function of the total number of dedicated protected lambda services.

$N = 30$, $L = 71.$, and $W = 160$.