



UNIVERSITEIT VAN AMSTERDAM

SNE

System and Network Engineering

On the Impact of Network Topology Aggregation in Multi-Domain Lightpath Provisioning

*Jeroen J. van der Ham, Paola Grosso, Freek Dijkstra, Arie Taal and
Cees T.A.M. de Laat*

October 11, 2010

Abstract

Dedicated network circuits (lightpaths) are an effective way to provide enhanced quality of service to demanding applications. Lightpaths are popular in the scientific community, where applications transport large quantities of data around the world for distributed visualisations or computations. Many research networks support this service model internally and co-operate to extend the service beyond their network boundaries.

Inter-domain lightpath provisioning requires some degree of knowledge of topology data, but network operators do not always wish to exchange their full topology, either for security, business, or for scalability reasons.

We present in this article a simulation study on the impact of topology aggregation for multi-domain lightpath provisioning. In particular we focus on the consequences of aggregation in the path finding process. Our work shows that a full mesh representation of the border nodes provides the best results among the various aggregation strategies. Our study uses realistic topologies for hybrid optical networks and this leads to results different from previous studies.

1 Introduction

Hybrid optical network architectures have become popular in many research and education networks[1]. Two different levels of service are available on the same physical infrastructure: the majority of users use the routed IP services; bandwidth hungry, delay sensitive applications utilise dedicated circuits. The dedicated circuits are called lightpaths. The advantage for the provider is in the separation of regular traffic from more QoS sensitive data streams[2].

In a single network domain the operators are free to choose management and control plane to create the circuits. Multi-domain lightpaths present instead many challenges, as different management and control planes have to interoperate[3]. The exchange of network topology information is an essential component in this process; in [4] we have presented a distributed information model to address this issue. Still, network operators do not always wish to share their full topology, either for security, business, or for scalability reasons. Topology abstraction and topology aggregation are an alternative to full information exchanges.

Our research has focused on the use of aggregation for the description of hybrid networks. There are several different ways of aggregating network topologies, more on this in section 2. However it is unclear what the performance difference is between these different approaches. Using emulations we have attempted to quantify the impact of aggregation on inter-domain path finding. This article presents our results.

2 Related work

2.1 Topology aggregation approaches

Topology aggregation is not a new research topic. One of the first standards on the area of topology aggregation was in the Private Network-to-Network Interface (PNNI) Specification[5] of the ATM forum. For the PNNI hierarchical routing Lee[6] provided an overview of three aggregation methods: *Symmetric Node*, *Symmetric Star* and *Full Mesh*. We adopted this classification in our work.

In the *Symmetric Node* approach, also called *Simple Node*, the topology of a complete domain is replaced by a single node, which is directly connected to other domains (bottom-left of Fig. 1). A single metric is advertised for the connectivity through this node. This single parameter implies that all connectivity through the node is considered to be symmetrical.

In the *Symmetric Star* approach the topology of a domain is aggregated using a central node (bottom-centre of Fig. 1). The border nodes of the original topology and their inter-domain connections are preserved. The intra-domain connections are represented by virtual links, *spokes*, to

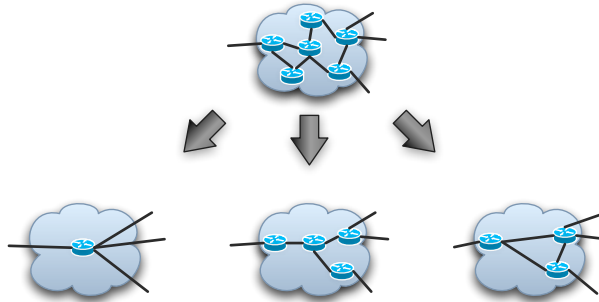


Figure 1: *Symmetric Node (left), Symmetric Star (center) and Full Mesh (right) aggregated topologies*

a virtual central node, the *nucleus*. All connectivity in this topology runs through the nucleus. This aggregation method is often referred to as *Star* aggregation. This is called a symmetric approach because a default definition for the properties of the spokes is used. All the spokes then have the same properties, unless explicitly specified otherwise.

The *Full Mesh* aggregation preserves the most information of the original topology (bottom-right of Fig. 1). Like in the Star aggregation, the border nodes are kept in the aggregated topology. Instead of a central node in the aggregated topology, there is a full mesh of connections between the border nodes. These aggregated connections between the border nodes can accurately describe the properties of the path through the domain, thus preserving the most important information for connections crossing the domain.

2.2 Performance evaluation

An aggregated topology hides details about the intra-domain connectivity; as a consequence inter-domain pathfinding on an aggregated topology is not always optimal. There are two immediate performance impacts. First, in an aggregated topology there is no discernible difference between a domain that has many internal hops and a domain with one hop. For this reason, the paths found in the aggregated topology may not be the shortest paths in the physical topology. Second, a path found in the aggregated topology sometimes does not map to a path in the physical topology due to the lack of available resources. This means that the path found in the aggregated topology is a false positive.

Evaluations of the performance and scalability of aggregation methods exist in the literature. We summarize here the results of three published studies on this topic: the work of Guo and Matta on a single emulated ATM network[7]; the work of Awerbuch et al. on several emulated topologies with

ATM networking[8] and the study on aggregated topologies in an optical network by Liu et al.[9, 3]

2.2.1 Performance Evaluation Study by Guo and Matta

Guo and Matta[7] examined the performance of aggregation methods in a single ATM topology, which they randomly divided in domains. They used two traffic workload schemes: a uniform workload, where source and destination pairs are uniformly distributed over the network; and a skewed workload, where some nodes are selected as destination for the majority of the connections.

Their conclusion was that under uniform workload Full Mesh and Star outperform Simple Node significantly; under skewed load the Simple Node performs better than or as well as Full Mesh and Star aggregations. In all cases the Star approach performs slightly worse than the Full Mesh as the former provides a less detailed view of the available bandwidth.

2.2.2 Performance Evaluation Study by Awerbuch et al.

Awerbuch et al. in [8] also compared aggregation schemas in specifically designed topologies, as well as randomly generated networks. They studied performance when using several aggregation methods, not only the ones proposed by Lee. They introduced two types of link metrics: links with constant cost functions and links with exponential cost functions, where the link cost increases exponentially as the available bandwidth decreases. To solve the problem of false positives they used *crank-back*, where information regarding this false-positive is then used in subsequent attempts to find a path. This also gives an indication for the set-up delay, because route recalculation when crank-backs occur is a time consuming task.

The results of the simulations confirm the authors' earlier theoretical work[10] that an exponential metric performs much better than a constant metric. In fact, even the worst aggregation strategy in the exponential metric simulation performs better than the best aggregation method in the constant metric simulation. Their results also show that the Star aggregation performs worse than other methods. It should be noted they did not consider the Single Node approach.

2.2.3 Aggregated Topologies in Optical Networks

Liu et al.[9, 3] have applied the Simple Node and Full Mesh aggregations to inter-domain WDM networks. This provides us with interesting results as aggregations had been extensively studied in ATM networks, but not in optical networks. In both cases the authors used a wavelength availability vector to represent the available wavelengths. The aggregation strategies are tested using two different lightpath provisioning strategies. They use the

term *transparent* for lightpaths that have the same end-to-end wavelength, and *translucent* for lightpaths that use different wavelengths using optical-electrical-optical (OEO) conversions.

The authors show results of their simulation on a topology with 9 domains and 19 inter-domain links, using 8 or 16 wavelengths. For transparent lightpaths Full Mesh performs better than Single Node aggregation. In the translucent case, there is still a difference between the two aggregation schemes, but the difference is smaller than in the transparent case.

3 Simulation setup

We have implemented a simulation to test the different aggregation methods. There are two main steps in the simulation: domain generation, and path finding in aggregated topology.

3.1 Domain generation

There are a large number of ways to generate random graphs. Barabási and Albert have shown in 1999[11] that many complex networks, exhibit a scale-free property. They determined that the probability $P(k)$ that a vertex in the network interacts with k other vertices decays as a power law, following $P(k) \sim k^{-\gamma}$. The value of γ varies with different types of graphs, but is usually $2 \leq \gamma \leq 3$. In case of the BGP router network, the value of $\gamma \sim 2$.

It is still unclear whether current hybrid optical networks are scale-free. The networks are too small and nodes have too small degrees to come to a definitive conclusion. We believe that as optical networks grow larger, they will also follow the power-law distribution. For this reason we chose the Barabási-Albert algorithm as implemented in the NetworkX Python module [12] to generate the graphs, and follow the procedure outlined in Alg. 1.

Generate D domains
 Generate N nodes in each d that belongs to D
 Create list of inter-domain node pairs: $l = (n_i, n_j)$ where n_i belongs to d_i and n_j belongs to d_j

Algorithm 1: Domain generation with Barabási-Albert

This generation method yields a different γ than the plain Barabási-Albert algorithm. Analysis of the results of this generation method shows that γ averages to ~ 2.3 , which is comparable to BGP networks.

In total we have $(D \cdot N) \cdot ((D - 1) \cdot N) \cdot \frac{1}{2}$ number of pairs. Before each run we shuffle this pair-list.

3.2 Path finding using aggregation

In this step we create a network emulation of the graph G . Starting from the full graph we create a Full Mesh aggregation (see Alg. 2), a Star aggregation (see Alg. 3) and a Simple Node aggregation (see Alg. 4).

```
for each domain  $d$  do
  for each boundary node  $b$  do
    for each boundary node  $b'$  do
      Search for a path from  $b$  to  $b'$ 
      if path exists then
        Add connection  $(b, b')$  to aggregated view
      end if
    end for
  end for
end for
Add inter-domain connections to aggregated view
```

Algorithm 2: Full mesh aggregation

```
for each domain  $d$  do
  Create virtual device nucleus  $d_n$ 
  for each boundary node  $b$  in  $d$  do
    if  $b$  has intra-domain connectivity then
      Add connection from  $b$  to  $d_n$  to aggregated view
    end if
  end for
end for
Add inter-domain connections to aggregated view
```

Algorithm 3: Star aggregation

Once we have generated the aggregated topologies, our path finding algorithm proceeds in the same manner independently of the aggregation strategy. The pseudo-code in Alg. 5 illustrates our method.

In all cases we use a standard Dijkstra's shortest path algorithm. As a baseline we perform pathfinding on the graph using complete information of the graph, that is, we work with the *full graph*. This is the ideal case, to which all aggregation methods should approximate.

Line 2 of Alg. 5 shows that for each pathfinding attempt on the pair of endpoints (x, y) , we add x and y to the aggregated graph. We do this according to Alg. 6.

Line 12 of Alg. 5 shows that we update the intra-domain topology by removing used links. This extra operation make sense only in the Full Mesh

```

for each domain  $d$  do
    Create a single node that represents the domain
end for
for each inter-domain connection in full topology do
    Determine domain endpoints of connection,  $d_1$  and  $d_2$ 
    Select single nodes  $s_1$  and  $s_2$  representing  $d_1$  and  $d_2$ 
    Add connection between  $s_1$  and  $s_2$  to aggregated view
end for

```

Algorithm 4: Single Node aggregation

```

1: for each pair  $(x, y)$  in  $l$  do
2:   Add  $x$  and  $y$  to the aggregated view
3:   Search a path between  $x$  and  $y$  in the aggregated view:
4:   if path exists then
5:     Translate path to full-topology view full_path
6:     if full_path is available then
7:       Record result
8:     else
9:       Record false positive
10:    end if
11:   end if
12:   [Update intra-domain topology by removing used links]
13:   Remove  $x$  and  $y$  from the aggregated view
14: end for

```

Algorithm 5: Path finding strategy

```

for each  $(x, y)$  do
  if Aggregation strategy is Full Mesh then
    Search for a path between  $x$  and all boundary nodes in  $d_x$ 
    if path exists then
      Add connection to aggregated view
    end if
    Repeat for  $y$  in  $d_y$ 
  end if
  if Aggregation strategy is Star then
    Add connection from  $x$  to nucleus  $d_n x$ 
    Add connection from  $y$  to nucleus  $d_n y$ 
  end if
  if Aggregation graph is Single Node then
    Add connection from  $x$  to single node for  $d_x$ 
    Add connection from  $y$  to single node for  $d_y$ 
  end if
end for

```

Algorithm 6: Addition of end nodes x and y to aggregated graph

aggregation strategy. We should point out that the Full Mesh without updates is very similar to the Simple Node method. The Full Mesh without updates shows the edge nodes, with a full mesh between them. The Simple Node graph only shows a single point, implicitly assuming full connectivity between all its inter-domain interfaces. Regardless of whether intra-domain updates occurs, all of the aggregation methods do update the inter-domain connectivity. In our experience the mean time between lightpath requests can be measured in the order of days or even weeks, so in this study we do not take the propagation delays into account.

4 Results

Each time a path is found, we record the result (success or false-positive), length of that path, and the new resource usage of the network. From each simulation run we have a large set of results (10^4 data points per aggregation method).

The length of the paths will change as the network is gradually filled up. Initially path lengths are fairly stable, as the network is still empty and almost any path will succeed. Paths will start at the average path length in that network, and gradually increase. As the network starts to fill up, most requests will still succeed, but the path length peaks as longer and longer detours are taken. This grows until the network becomes nearly saturated, meaning that large parts of the network are in use. Then only

small disconnected parts of the network remain available, and the chance of success depends on the distance in the network. The path lengths will gradually decrease to the minimum, i.e. paths between neighbours.

To show the combined effects of the path length and the success rate we examine the development of inter-domain resource usage over the successive requests. The inter-domain resource usage is defined as the fraction of links between domains that have been used in successful requests. We have set this against an index describing the number of path requests that have been submitted to the network so far. This index is normalized to the complete graph-size, also called the Relative Index below.

There are two different behaviours detectable over successive requests, the initial increase in path lengths, and the slow decrease once the network reaches a saturation point, as shown in figure 2. Therefore we use two different functions for fitting to the results. We determine the split by the path length peak. The peak and everything before is the first part, and afterwards is the second part.

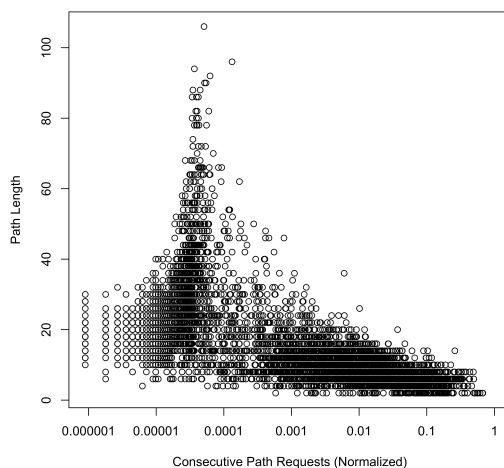


Figure 2: A scatterplot showing the path length distribution in full view pathfinding

Initially there is a constant success rate, and the path length increases linearly, so we use a linear function starting at zero as shown in equation 1.

$$InterDomainUsage = A \cdot RelativeIndex \quad (1)$$

In the second section the behaviour is dictated by a decreasing success-rate and path length. This follows a slow logarithmic growth in inter-domain resource usage. So we fit the results using the function shown in equation 2.

$$\text{InterDomainUsage} = A \cdot \log(\text{RelativeIndex}) + B \quad (2)$$

In figures 3 and 4 we show the fits for the results on the graphs with $(d = 150, n = 5)$. Tables 1 and 2 show the fitted values, along with their errors, and the explained variance.

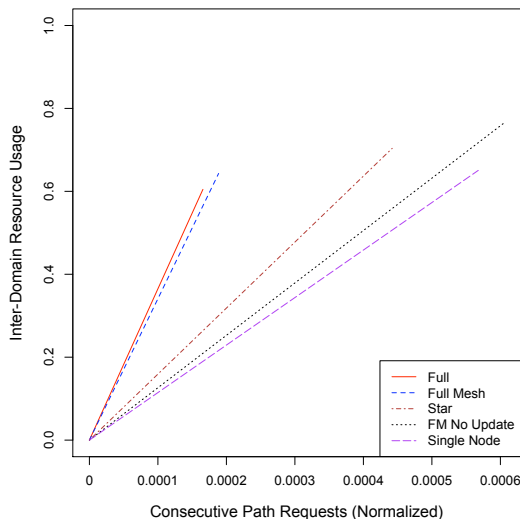


Figure 3: Fitted functions for the initial linear growth phase

Strategy	A	σ	R ²
Full	3649	7	0.99
Full Mesh	3409	6	0.99
FM no updates	1263	6	0.90
Star	1591	7	0.93
Single Node	1145	6	0.91

Table 1: Fitted values in the initial linear growth phase

The graph of the first section in figure 3 shows the initial linear growth. Recall that the boundary for the first section is determined by the peak in the path lengths. The fitted graphs all end around 60% resource usage, however the more aggregated the longer it takes to get there. The same difference in performance continues in the logarithmic growth part of the fit. The Full Mesh aggregation performs almost perfectly compared to the not aggregated case. The Star aggregation method shows a significantly lower performance, with the Full Mesh without updates just below it. The Single Node aggregation method clearly performs worst of all the aggregation methods.

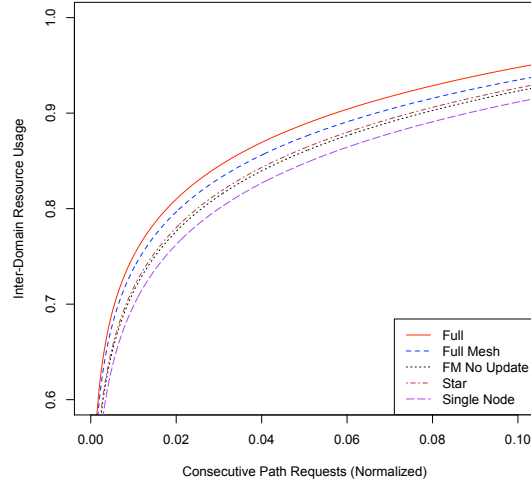


Figure 4: Fitted functions for the logarithmic growth phase

Strategy	variable	value	σ	\mathbf{R}^2
Full	A	0.0857	0.0002	0.91
	B	1.1451	0.0017	
Full Mesh	A	0.0858	0.0002	0.92
	B	1.1322	0.0015	
FM no updates	A	0.0912	0.0001	0.95
	B	1.1331	0.0011	
Star	A	0.0908	0.0002	0.94
	B	1.1355	0.0015	
Single Node	A	0.0926	0.0001	0.96
	B	1.1250	0.0009	

Table 2: Fitted values in logarithmic growth phase

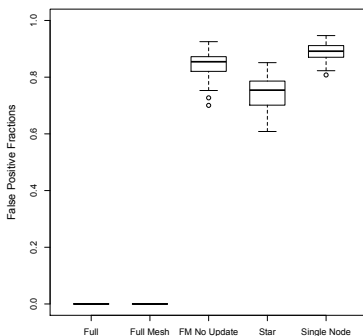


Figure 5: A box plot of false positive fractions for ($d = 150, n = 5$)

In figure 5 we show the fractions of false positives in a box plot. It should be noted that these false positives only occur in the logarithmic growth phase. None of the aggregation methods show false positives in their linear growth phase.

It is not possible to have false positives with the full view, nor with the Full Mesh view with updates. In the latter case, the update mechanism always makes the graph reflect the current availability in the network. The number of false positives is extremely high in the aggregation strategies without detailed intra-domain connectivity in the logarithmic growth phase. When the Single Node is used, only 1 in 10 attempts will result in an actual path. Even with the Star aggregation, which shows some intra-domain details, over 75% of the attempts is a false positive result. This clearly shows that once the network resources becomes less and less available, detailed knowledge of intra-domain connectivity is required in order to provide accurate results for inter-domain pathfinding.

5 Discussion

Our results clearly show that aggregation does indeed have an impact on the performance of inter-domain pathfinding. In the initial linear growth phase the Full Mesh aggregation strategy performs close to the Full View. The other aggregation strategies perform significantly worse.

This difference in performance becomes much smaller in the logarithmic growth phase, where the growth of inter-domain resource usage in all aggregation strategies is very similar. However in this phase the number of false positives in the aggregation methods without accurate intra-domain connectivity is extremely high. This means that once the network becomes reasonably filled, these aggregation strategies become almost unusable without a way of filtering out these false positives.

A possible way of using the information from false positives is by using crank-backs. This method of locally updating the view of the topology using false-positive information ultimately creates a similar view on the graph as the Full Mesh method does. The difference is that with crank-back the majority of the effort of creating the updated graph lies with the requester. The resulting graph is also not shared with the other domains. The load of pathfinding is then shifted from the domains, performing less updates, to the source, which has to perform the crank-backs. We have seen from the false-positive results that clients will very often have to perform these crank-backs in saturated networks.

An argument that is often used in favour of using aggregation is scaling: finding paths in large detailed graphs takes more time than finding a path in an aggregated graph. This argument fails to take the cost of constructing and updating the aggregated graph into account. In the case of the Full Mesh graph with updates, the cost of maintaining the graph is distributed over all the domains. The total distributed processing time is then higher than finding a path in a full detailed graph. The aggregated topologies are less hard to maintain in the Star, Simple Node, or Full Mesh without updates, but these views show a very large number of false-positives. To get reasonable performance in these strategies, very time consuming crank backs have to be used.

It is somewhat difficult to compare our results to the results of Awerbuch et al. since they use crank-backs. However, the general performance trends in their results are similar to ours. Full Mesh ('Complete' in their terminology) performs best, while both their Star aggregation methods show a slightly worse performance.

Comparing our results to the results of Guo and Matta, we see a significant difference in the performance of the Star aggregation. In their study the Star performs almost equally with the Full Mesh aggregation, while both in Awerbuch et al. and our results the Star aggregation performs significantly worse. Unfortunately, we cannot reproduce their results, even when using their topology we see a significant difference between the Full Mesh and Star aggregations.

We have presented our results to them, and they responded that the difference is probably caused by the different approaches in our simulations. In their simulations they incorporate a delay between topology updates communicated to other domains. In our simulations there is no delay, however, we have also shown our results of the Full Mesh without updating. FM without updates performs slightly worse than Star in our case. However, when we also run the Star aggregation without updates, it performs significantly worse than FM without updates.

6 Future work

A direct comparison with the results of Liu et al. is not possible, because we have not allowed partial use of links as they do when multiple wavelengths runs on the same connection. But such a comparison is very interesting given that DWDM is often the core technology in hybrid networks.

In general, our results would benefit if we consider the various technologies used in hybrid networks. This means including information over the various networks layers, and making this information available to the path finding algorithm. Work on multi-layer pathfinding[13] exists, but the algorithm is based on full topology information. An open issue is defining aggregation strategies for multi-layer topologies. This is far from trivial: besides aggregation of the connectivity information, the encoding and the adaptation capabilities must also be considered. We intend to perform this research next.

Another open issue is mixed aggregation strategies. Our simulations used the same aggregation strategy for all domains. In practice different domains will make different choices for topology aggregation. We intend to investigate what effect mixed aggregation have on the performance of the inter-domain network pathfinding.

7 Conclusion

Hybrid networks provide a new model for support of high-demand applications, and inter-domain path finding is an essential component in building world-wide lightpaths. Path finding algorithms need information on the available network resources; the level of detail provided by each domain impacts the performance and outcome of a path finding search.

In this article we have examined what kind of impact aggregation has on the performance of inter-domain pathfinding. We have described different aggregation methods, and performed simulations to test and quantify this impact. Our study differs from existing studies as it uses realistic topologies for hybrid optical networks, with multiple domains and inter-domain links. We can conclude that a full mesh representation of the full topology provides the best results among the various aggregation strategies. False positive and false negative paths can be avoided by updating the topology information in the aggregated views, but this increases the processing required at each domain.

Acknowledgment

The authors would like to thank the Dutch national research and education network SURFnet and the Dutch National Research Foundation (NWO) for

their financial support in this research.

References

- [1] L. Smarr, T. DeFanti, M. Brown, C. de Laat, Special section: iGrid 2005: The global lambda integrated facility, *Future Generation Computer Systems* 22 (8) (2006) 849–851.
- [2] C. de Laat, E. Radius, S. Wallace, The rationale of the current optical networking initiatives, *Future Generation Computer Systems* 19 (6) (2003) 999–1008. doi:10.1016/S0167-739X(03)00077-3.
URL <http://www.sciencedirect.com/science/article/B6V06-48V83MF-5/2/d8aac1d72ec497da8c83c4a07dfec0c>
- [3] N. Ghani, Q. Liu, D. Benhaddou, N. S. V. Rao, T. Lehman, Multidomain optical networks: issues and challenges - control plane design in multidomain/multilayer optical networks, *Communications Magazine, IEEE* 46 (6) (2008) 78–87. doi:10.1109/MCOM.2008.4539470.
URL <http://dx.doi.org/10.1109/MCOM.2008.4539470>
- [4] J. van der Ham, F. Dijkstra, P. Grosso, R. van der Pol, A. Toonk, C. de Laat, A distributed topology information system for optical networks based on the semantic web, *Optical Switching and Networking* 5 (2-3) (2008) 85–93.
URL <http://www.science.uva.nl/~vdham/research/publications/0703-ApplicationsOfNDL.pdf>
- [5] Private network-network interface specification, Tech. rep., ATM Forum (1996).
URL <http://www.ipmplsforum.org/ftp/pub/approved-specs/af-pnni-0055.001.pdf>
- [6] W. C. Lee, Topology aggregation for hierarchical routing in atm networks, *SIGCOMM Computer Communications Review* 25 (2) (1995) 82–92. doi:10.1145/210613.210625.
- [7] L. Guo, I. Matta, On state aggregation for scalable QoS routing, in: *Proceedings of the ATM Workshop*, Vol. 6, 1998, pp. 306–314.
URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.8296>
- [8] B. Awerbuch, Y. Du, B. Khan, Y. Shavitt, Routing through networks with hierarchical topology aggregation, in: *Third IEEE Symposium on Computers and Communications (ISCC)*, 1998, pp. 406–412. doi:10.1109/ISCC.1998.702556.

- [9] Q. Liu, M. A. Kök, N. Ghani, A. Gumaste, Hierarchical routing in multi-domain optical networks, *Computer Communications* 30 (1) (2006) 122–131.
- [10] B. Awerbuch, Y. Azar, S. Plotkin, Throughput-competitive on-line routing, in: *SFCS '93: Proceedings of the IEEE 34th Annual Foundations of Computer Science*, IEEE Computer Society, 1993, pp. 32–40. doi:10.1109/SFCS.1993.366884.
- [11] A. L. Barabási, R. Albert, Emergence of scaling in random networks, *Science* 286 (5439) (1999) 509 – 512.
- [12] A. Hagberg, D. Schult, P. Swart, Networkx python package (version 0.36).
URL <https://networkx.lanl.gov/wiki>
- [13] F. Kuipers, F. Dijkstra, Path selection in multi-layer networks, *Computer Communications* 32 (1) (2008) 78 – 85. doi:10.1016/j.comcom.2008.09.026.
URL <http://staff.science.uva.nl/~fdijkstr/publications/multilayer-pathselection.pdf>