Open Marketplace and Service Orchestration for Virtual Optical Networks

(Invited Paper)

Shireesh Bhat*, George N. Rouskas†‡
*ECE Department, University of California Santa Barbara, USA
†Department of Computer Science, North Carolina State University, USA
‡Department of Computer Science, King Abdulaziz University, Saudi Arabia

***Abstract*—A key challenge in multi-vendor heterogeneous virtual optical networks is providing transparent access to network resources and virtual functions in a manner that enables users to combine them appropriately into meaningful end-to-end services. In this paper, we present a solution that consists of two components: an open marketplace where vendors and users of network resources and functions meet to establish economic relationships; and a planning service for creating end-to-end communication services from functional building blocks available in the marketplace. We also discuss algorithms for tackling variants of the network service orchestration problem.**

I. INTRODUCTION

In recent years the focus in the networking field has been on developing modular systems and moving away from the monolithic designs of the past. Specifically, network virtualization [1] decouples service functionality from the underlying resources (including network, compute, and storage) that are involved in delivering the service. Consequently, virtualization makes it possible to deliver end-to-end communication services that are composed from functional building blocks that (1) may be available at various locations strategically dispersed across the network, and (2) may be offered by different providers. By allowing for multiple service providers to coexist on the same physical network substrate but separated by a virtualization layer, it is expected that network virtualization will lead to increased provider competition, more innovation, and more options/choices for users, as providers develop value-added services within their virtual network to stand out from the competition.

In general, network virtualization has been concerned with higher layers of the networking stack, and for the most part it has not touched the physical layer. In other words, the optical layer has typically been considered as a "black box:" sequences of bits are delivered to it for transmission, without the higher layers being aware of exactly how the transmission is accomplished. This separation of concerns imposed by the layering principle has allowed the development of upper layer protocols and services that are independent of the physical channel characteristics, but it has now become too restrictive as it prevents protocols or applications from taking advantage of additional functionalities that are increasingly available at the optical layer. In particular, in the past few years we have witnessed the development of optical layer devices that are *intelligent*, *self-aware*, and *programmable*, in that they can sense or measure their own characteristics and performance, and their behavior can be altered through software control.

The capabilities and functionality of these devices must somehow be exposed to higher layer applications and protocols, hence current network architectures cannot capture the full potential of the optical layer. For instance, the optical substrate increasingly employs various optical monitors and sensors, variable optical attenuators, bandwidth-variable transponders, distance-adaptive modulation, amplifiers and other impairment compensation devices. The monitoring and sensing devices are capable of measuring loss, polarization mode dispersion (PMD), or other signal impairments; based on this information, it should then be possible to use the appropriate impairment compensation to deliver the required signal quality to the application/user on demand. But such a solution cannot be accomplished within the current architecture, and has to be engineered outside of it separately for each application and impairment type; clearly, this is not an efficient or scalable approach. Reconfigurable optical add-drop multiplexers (ROADMs), flexible spectrum selective switches, and optical splitters with tunable fanout (for optical multicast) are additional examples of currently available devices whose behavior can be programmed according to the wishes of higher layer protocols. Based on current research trends one may anticipate further innovation in this field leading to the development of other sophisticated devices with programmable functionality that may be tailored to address specific requirements of higher applications.

Making this functionality available for delivering customized higher level end-to-end services to users presents two challenges. First, the device capabilities must be exposed to higher layer protocols, applications, and users in a manner that enables users and providers to form economic relationships around virtual optical network services that make use of these capabilities. Second, general purpose planning tools must be developed to stitch together lower-level functional blocks into meaningful end-to-end services. Therefore, in Section II we present a marketplace for the discovery, creation, and exchange of network services, and in Section III we discuss high-level algorithms for the orchestration of optical network resources. We carry out an evaluation of the algorithms in Section IV and we conclude the paper in Section V.

978-3-903176-07-2

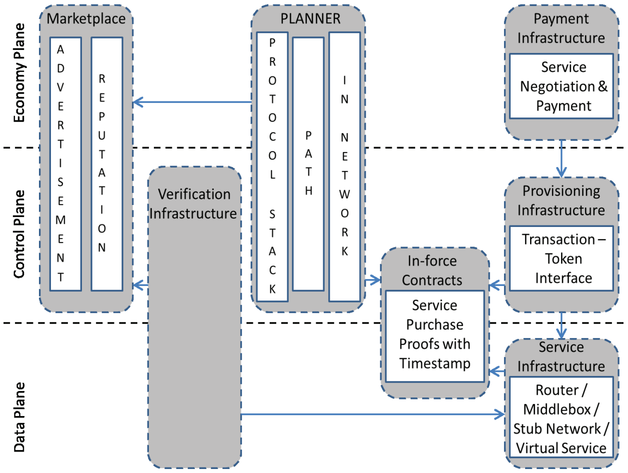

Fig. 1. A Marketplace of network services within the ChoiceNet architecture

II. A MARKETPLACE FOR NETWORK SERVICES

The Internet *edge* has thrived as an ecosystem where interactions among many stakeholders, acting as customers and/or providers of various hardware and software services for network, compute and storage functions, are mediated by economic considerations. However, this economic reality does not apply to any of the software or protocol interactions *within* the network, hence there is a lack of economic incentives for providing corresponding innovative services. To bridge this gap, as part of the ChoiceNet project [2], we introduced new mechanisms into the Internet architecture to enable an “economy plane” that allows the presentation of competing offerings for various networking services, the formation of contracts between users and providers, and determination of whether each provider meets its part of the contracts.

The ChoiceNet architecture, illustrated in Figure 1, introduces a platform for service providers to advertise their services and for customers to discover, negotiate and pay for these services. The main architectural component facilitating service advertisements is the “marketplace,” where service providers register their services and customers discover them via appropriate queries. Once a customer selects a service, further ChoiceNet interactions between customer and provider create a contract, with the customer receiving a token that is carried by its traffic as proof of purchase in order to receive the corresponding service. These customer/provider interactions constitute what we term the “economy plane” of ChoiceNet, shown on Figure 1 as separate from, but interacting with, the well-known control and data planes. We emphasize that the economy plane does *not* introduce new economic interactions; rather it empowers existing real-world interactions to take place (1) in-network (whereas today they take place outside the network architecture), and (2) at a range of short (e.g., in the order of a flow duration) to long time scales.

The marketplace is a virtual repository of network functions and services available to users. The repository provides interfaces for providers to list (advertise) their services, and for users (or their software agents) to obtain listings of service offerings that meet their requirements. ChoiceNet’s interfaces enable entities to realize complex service models where an entity may act as a provider to some customers and at the same time act as a customer to some providers. This feature is essential in a virtual network architecture as it enables service providers who lease physical network resources from infrastructure providers, in turn to lease their virtual network resources to other service providers.

Service advertisements in the ChoiceNet marketplace and protocol interactions in the economy plane are semantically enriched [3], [4] to allow automated composition, thus the marketplace is more like an ontology than a directory. As a result, although the ChoiceNet marketplace was conceived and demonstrated within a packet-switched network context [5], it may readily accommodate the optical layer functions and services we discussed in the previous section. For instance, consider an optical multicast service that is offered by deploying optical splitters in various nodes across the network. The service description in this case would include the address of the locations where the splitters are present, the maximum fanout of each, the spectrum range over which the splitters operate, the power loss due to splitting, and other relevant information that an orchestration algorithm may take into account in formulating a multicast communication service.

Realistically, orchestrating a set of services in the marketplace to create a complete service for a customer is expected to be a complex task for all but the simplest cases, thus the task must be automated and performed by software agents. The planner module in the ChoiceNet architecture of Figure 1 interacts with the marketplace over the economy plane and employs specialized algorithms to orchestrate marketplace services into end-to-end communication services for users. We note that the architecture allows for multiple planning services to co-exist in competition to each other: each planning service may query the marketplace repository to obtain the available services and hence focus on innovation in the design of orchestration algorithms to enhance customer experience. The reader may have noticed the analogy with a real-world application, the travel industry, a model that has guided our design of the ChoiceNet architecture. In the travel industry, service providers include the airlines, hotels, and rental car companies, whereas travel sites such as Orbitz or Priceline operate planning and orchestration services. These sites take as input traveler preferences and construct itineraries to ensure that users may access seamlessly all the services acquired across the various flight, accommodation, and car rental providers.

We assume that planners represent the services available in the marketplace in a graph format, such that service orchestration may be carried out using appropriately designed graph algorithms. We expect that such a graph will be highly dynamic as it must be updated whenever a user acquires or releases services. Also, the planner for a marketplace of virtual optical network services must consider offerings from multiple providers, including virtual operators who may lease resources from the same physical infrastructure or recursively lease services from other providers. Returning to the travel analogy, this is similar to a planner taking into consideration competing flights from multiple airlines between pairs of cities, as well as multiple hotels or rental car agencies within a city. Consequently, the graph of marketplace services is a superset of the underlying network topology. Specifically, nodes and edges in the services graph represent virtual entities rather than physical ones: a physical node may include multiple virtual nodes, each virtual node operated by a different service provider deploying a variety of network service instances. The graph may also include parallel edges between nodes that represent competing path services. Such a topology may be considerably larger than the underlying physical network topology, hence orchestration algorithms must scale to large graph sizes.

III. SERVICE ORCHESTRATION

As we mentioned in the previous section, we expect that the deployment of marketplaces for network services will lead to innovation in the design and application of orchestration algorithms for the delivery of customized end-to-end communication services in virtual optical networks. In turn, the availability of planners that operate in short time scales (i.e., on par with the setting up of a service) is likely to generate further customer interest in specialized services, which will motivate virtual network providers to invest in novel services and more sophisticated orchestration algorithms to differentiate from the competition, creating a virtuous cycle similar to the one we have witnessed unfold at the edge of the network in the past twenty five years.

Broadly speaking, upon receiving a request from a user (customer), a planner must carry out three tasks as part of the service orchestration process [6], [7]:

- *Service Selection:* determine the set of virtual network services to satisfy the user request;
- *Service Ordering:* determine the order in which the selected services must be applied to the user's traffic; and
- *Service Concatenation:* construct path(s) from the source node to the destination node(s) that visit virtual nodes where instances of the selected services are deployed in the order determined by the service ordering step.

In previous work, we have considered the service orchestration problem in contexts where the three subproblems above are pairwise decoupled and may be carried out sequentially in the given order. In such situations, the service concatenation step will involve general algorithms that may applied to a broad set of services, as we discuss next.

Let us assume that multiple instances of each service k are deployed at various virtual nodes across the network, possibly operated by different service providers. Also let S_k denote the set of nodes where instances of service k are deployed. In earlier work [8] we considered the following general service orchestration problem for a user request that requires $K \geq 1$ services to be applied in a given order:

> Given the graph representing the union of the virtual network topologies represented in the marketplace, a source node s, a destination node d, and an ordering of K node sets $S_1, S_2, \ldots, S_K$, construct a path of minimum cost from s to d that visits one node in each set $S_k, k = 1, \cdots, K$, in the given order.

The above problem is equivalent to the shortest path tour problem (SPTP) that was first studied in a different context more than forty years ago [9], [10]. A shortest path tour is a path of minimum cost from s to d constructed as the concatenation of the $K+1$ path segments $[s, n_1], [n_1, n_2], \cdots, [n_K, d]$, where $n_k \in S_k, k = 1, \cdots, K$, and each segment may include nodes other than its two endpoints.

SPTP is solvable in polynomial time, but as we discussed in [8], earlier algorithms were designed for only certain classes of graphs (either sparse graphs or small dense graphs), and do not scale to graph instances we expect will arise in representing marketplaces of virtual network services. We also developed a new algorithm by introducing several novel modifications to Dijkstra's algorithm to construct the shortest path tour efficiently. This new algorithm, which we call depth-first tour search (DFTS), scales to graphs with thousands of nodes and large nodal degrees, and is appropriate for real-time service orchestration applications.

The DFTS algorithm, as well as earlier algorithms for the SPTP problem were developed for packet-switched networks, but they may certainly be applied in the context of virtual optical networks offering a range of services from the physical layer (including the ones we listed in Section I) to the application layer (as we have discussed in [6], [8]). For instance, consider a user request that requires services to be applied directly to the optical signal (e.g., amplification, dispersion compensation, etc.) carrying the user traffic, as well as transformation services (e.g., transcoding of application data, encryption or decryption, and more) to be applied to the data carried by the signal. As long as an ordered set of services is provided to the service concatenation step, then appropriate algorithms for the SPTP problem may be used to construct minimum cost paths that include nodes where the services are offered.

Often, however, the three subproblems of service orchestration (i.e., service selection, ordering, and concatenation), are not decoupled, hence solving them sequentially may not lead to an overall optimal solution (path) or even a feasible one. This may be especially true when optical layer services are part of the mix, due to cross-layer dependencies. For instance, for a given quality-of-service requested by the user, the selection subproblem may need to coordinate with the concatenation subproblem so as to take into consideration the length and other properties of the candidate optical path(s) in order to determine the modulation format or spectrum of the signal, or whether to include impairment compensation services. Although there has been considerable research in cross-layer optical network design [11], including routing algorithms that take into account physical layer impairments, to the best of our knowledge, the general service concatenation problem we defined above has not been studied when the three subproblems are tightly coupled.

A solution to the general service orchestration problem is outside the scope of this work and is the subject of ongoing research in our group. Nevertheless, we expect that more restricted variants of the problem will have applications in specific contexts. In this paper, we consider such a variant that arises in virtual networks offering multicast services at the optical layer. An optical layer multicast service may be used to implement point-to-multipoint connections directly at the physical layer by employing optical splitters that divide the power of an input signal into several output signals [12]. Let m denote the number of distinct destination nodes to which the signal must be delivered. Then, a multicast service with a fanout of at least m must be included in the service selection step of the orchestration process, along with any other services needed to satisfy the user request.

We consider a special case of the service orchestration problem wherein the service selection subproblem is independent of the other two subproblems, but the service ordering and service concatenation problems are coupled with respect to the order of the multicast service. More specifically, let $K, K > 1$, be the number of services, including the multicast service, determined by the selection step. Also assume that the relative order of the $K-1$ services other than the multicast service has been decided (i.e., it remains fixed and is not subject to optimization), but that the multicast service may be placed in any position in that relative ordering. For instance, amplification may take place before or after splitting the optical signal, keeping in mind that in the latter case, the amplification service must be applied to all m output signals. Similarly for higher layer services, since, say, encryption or transcoding may be applied to the original traffic stream or the m streams produced as the result of splitting.

The problem of finding the shortest path tour from source s to the m destination nodes is a generalization of the SPTP problem we defined above, and we refer to it as the point-to-multipoint SPTP (P2MP-SPTP). Consider now a special case of the problem whereby the multicast service is placed as the k-th service in the ordering, $1 \leq k \leq K$. Recall that S_k is the set of nodes where the multicast service is offered, and let $|S_k| = L \geq 1|$. Further, let $S_k = \{n_1, \cdots, n_L\}$. We may obtain an optimal solution to this special case by following these steps:

1) Initialize $i = 1$.
2) Set $S'_k = \{n_i\}$.
    2a) Solve the SPTP problem from s to n_i with input $S_1, \cdots, S'_k$.
    2b) Solve the SPTP problem from n_i to the m destination nodes with input $S_{k+1}, \cdots, S_K$.
    2c) Concatenate the two tours to obtain the tour from s to the m destinations, and record its cost.
3) Increment i and repeat Step 2 while $i \leq L$.
4) Select among the L tours constructed the one with the minimum cost.

Step 2a ensures that the final tour consists of a single path from s to some node $n \in S_k$ where the multicast service is applied to split the input optical signal into m output signals. Performing Step 2 for each node in set S_k guarantees that the shortest tour is found in the last step.

We may now obtain an optimal solution to the original problem by repeating the above algorithm K times, each time with the multicast service as service $S_i, i = 1, \cdots, K$, in the order of services, and selecting the best overall solution. Note that the worst-case running time complexity of the DFTS algorithm we presented in [8] is $O(KE \log N)$, where E and N represent the number of edges and nodes, respectively, in the underlying graph. Therefore, the complexity of the above approach is $O(LK^2 E \log N)$, which for moderate values of L and K (e.g., in the order of 10-20) would be reasonable and appropriate for online operation. In particular, our experimental evaluation of DFTS has shown that the algorithm completes in well under one second even for dense graphs with up to $N = 5,000$ nodes. Therefore, even with the additional $O(LK)$ factor, the above algorithm for the P2MP-SPTP problem may be used at the time scales appropriate for setting up end-to-end flows in real time.

Consider now the general case of the joint service ordering and concatenation problem, and for simplicity assume point-to-point communication only, i.e., a single source s and a single destination d. A straightfoward approach to solving this problem would be to solve each of the $K!$ SPTP problems that arise for each possible permutation of the K selected services. Furthermore, not all $K!$ permutations may be valid, resulting in a smaller solution space for the original problem: for instance, note that encryption must precede decryption or that transcoding must precede encryption. Nevertheless, for larger values of K, enumerating all valid permutations of services is expected to be computationally infeasible, especially for applications that require results in real time. Further research is necessary to determine the complexity of this problem and to derive polynomial-time algorithms, perhaps by extending existing SPTP algorithms, including the DFTS algorithm we presented in [8].

As a final note, we conjecture that the most general service orchestration problem whereby all three subproblems (service selection, ordering, and concatenation) have to be solved jointly is computationally intractable. Even if the conjecture is true, efficient algorithms may exist under certain simplifying assumptions that may hold in practice. Such algorithms are essential so as to account for the cross-layer dependencies inherent in the delivery of end-to-end communication services in virtual networks with programmable optical layer capabilities. Therefore, we consider this an important research direction and one that may readily build upon the insights from recent and ongoing research in multilayer optical network design.

IV. NUMERICAL RESULTS

We evaluate our algorithm on random graphs generated using BRITE [13], a universal topology generator. We obtained undirected graphs by configuring BRITE to generate AS-Level Barabasi models; we then converted these graphs into directed ones that we used in our experiments. In generating random instances for the P2MP-SPTP problem, we considered the following parameters and varied their values as described below:

- The number N of nodes in the graph was varied from 1000 to 5000 in increments of 1000.
- The average nodal degree Δ of the graph was set to an integer in the range $[2, 5]$.
- The number K of node sets in the tour was set to 4
- The number k for the relative order of the multicast service set took integer values in the interval $[1, 4]$
- The number M of nodes in the multicast service set was varied from 2 to 8 in multiples of 2. The number of nodes in the non-multicast service sets was set to 25
- The number of destination nodes (multicast streams) was set to 10.

There are 240 unique combinations of the values of parameters $N, \Delta, k,$ and M that we considered in our experiments (refer to the top of this section). In Table I, we list the actual running time of our algorithm, for problem instances generated with each of these 240 parameter value combinations. Each entry in the table is the average running time over 1,000 problem instances generated from the stated values of the parameters. All experiments were performed on a High Performance cluster that included Dual Intel X5650 six core processors, with 48GB and infiniband interconnect.

We make the following observations:

- The running time increases linearly as a function of M when $N, \Delta,$ and k are kept fixed.
- The running time increases linearly as a function of $ElogN$ when $M, \Delta,$ and k are kept fixed.
- The running time increases slower than linearly as a function of Δ, when
- When $N, M,$ and Δ are kept fixed, we observe that the relative order of the multicast service set produces some interesting results. In many instances when k is 1, we observe a marginally higher running time compared to the rest of the cases, but we do not see this pattern for all the instances. We infer that the inherent graph characteristics (N, E, Δ) influence k since we divide the P2MP-SPTP problems into multiple SPTP sub-problems and the value of k determines the size of the SPTP sub-problems.

To the best of our knowledge, this is the first work to address the P2MP-SPTP problem and develop an algorithm to solve it efficiently. We hope that this paper serves as a reference and paves the way for further investigation of the P2MP-SPTP problem.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under Grant number 1111088. We also acknowledge support from the Center for Scientific Computing from the CNSI, MRL: an NSF MRSEC (DMR-1720256) and NSF CNS-0960316.

V. CONCLUDING REMARKS

We envision virtual optical networks that empower end users to make informed choices in selecting among network services offered by competing providers within an ecosystem that promotes and rewards innovation. A marketplace that serves as repository of service building blocks and the meeting ground between customers and providers is the first component of such a vision. The second component consists of sophisticated orchestration algorithms that add value to the user experience by creating highly customized end-to-end communication services from the existing building blocks. We consider the development of orchestration algorithms that take into account cross-layer dependencies as a fruitful area of research for the optical networking community.

REFERENCES

[1] A. Wang, M. Iyer, R. Dutta, G. N. Rouskas, and I. Baldine. Network virtualization: Technologies, perspectives and frontiers. *IEEE/OSA Journal of Lightwave Technology*, 31(4):523–537, February 2013.

[2] T. Wolf, J. Griffioen, K. Calvert, R. Dutta, G. N. Rouskas, I. Baldin, and A. Nagurney. ChoiceNet: Toward an economy plane for the Internet. *ACM SIGCOMM Computer Communication Review*, 44(3):58–65, July 2014.

[3] S. Bhat, R. Udechukwu, R. Dutta, and G. N. Rouskas. On service composition algorithms for open marketplaces of network services. In *Proceedings of EuCNC*, June 2017.

[4] R. Udechukwu, S. Bhat, R. Dutta, and G. N. Rouskas. Language of choice: On embedding choice-related semantics in a realizable protocol. In *Proceedings of IEEE Sarnoff Symposium*, September 2016.

[5] S. Bhat, R. Udechukwu, R. Dutta, and G. N. Rouskas. Inception to application: A GENI based prototype of an open marketplace for network services. In *Proceedings of IEEE INFOCOM Workshops*, April 2016.

[6] S. Bhat and G. N. Rouskas. On routing algorithms for open marketplaces of path services. In *Proceedings of IEEE ICC 2016*, May 2016.

[7] X. Huang, S. Shanbhag, and T. Wolf. Automated service composition and routing in networks with data-path services. In *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, pages 1–8, Aug 2010.

[8] S. Bhat and G. N. Rouskas. Service-concatenation routing with applications to network functions virtualization. In *Proceedings of ICCCN 2017*, August 2017.

[9] C. P. Bajaj. Some constrained shortest-route problems. *Unternehmensforschung*, 15(1):287–301, 1971.

[10] A. Kershenbaum, W. Hsieh, and B. Golden. Constrained routing in large sparse networks. In *IEEE International Conference on Communications," pp. 38.14-38.18, Philadelphia, PA*, 1976.

[11] S. Subramaniam, M. Brandt-Pearce, and C. Vijaya Saradhi (Eds.). *Cross-Layer Design in Optical Networks.* Springer, 2013.

[12] G. N. Rouskas. Optical layer multicast: Rationale, building blocks, and challenges. *IEEE Network*, 17(1):60–65, January/February 2003.

[13] A. Medina, I. Matta, and J. Byers. Brite: A flexible generator of internet topologies. Technical report, Boston University, Boston, MA, USA, 2000.

TABLE I

AVERAGE RUNNING TIME (IN SECONDS) OF OUR ALGORITHM TO SOLVE P2MP-SPTP

		$k = 1$			$k = 2$			$k = 3$			$k = 4$		
N	Δ	$M = 2$	$M = 4$	$M = 8$	$M = 2$	$M = 4$	$M = 8$	$M = 2$	$M = 4$	$M = 8$	$M = 2$	$M = 4$	$M = 8$
1000	2	0.04324	0.08458	0.17306	0.0414	0.08166	0.16584	0.04056	0.08141	0.16102	0.0405	0.0797	0.16128
	3	0.0501	0.09792	0.19795	0.04751	0.09394	0.18942	0.04681	0.09387	0.18535	0.04676	0.09287	0.18596
	4	0.06994	0.13483	0.27256	0.06641	0.13228	0.26554	0.06552	0.13167	0.26078	0.06573	0.1303	0.26213
	5	0.07725	0.1528	0.30158	0.07404	0.147	0.29484	0.07389	0.14825	0.29393	0.07408	0.14719	0.29487
2000	2	0.17413	0.34667	0.68324	0.17126	0.33195	0.65452	0.16322	0.3212	0.64251	0.16298	0.32105	0.63134
	3	0.21837	0.43511	0.86546	0.21683	0.42536	0.84286	0.21247	0.4254	0.85354	0.21541	0.4222	0.83347
	4	0.26384	0.52766	1.04008	0.26234	0.51041	1.01864	0.25425	0.51412	1.01286	0.25942	0.50435	1.01448
	5	0.3053	0.60819	1.20743	0.30137	0.5911	1.17768	0.29254	0.5898	1.15671	0.29487	0.57646	1.16156
3000	2	0.4034	0.80138	1.58029	0.39145	0.77083	1.57162	0.39127	0.78169	1.55459	0.39346	0.76178	1.53388
	3	0.50435	0.98411	1.95159	0.48306	0.98064	1.94662	0.48391	0.96243	1.92723	0.488	0.9444	1.90324
	4	0.54207	1.11959	2.22108	0.55442	1.09921	2.21439	0.55444	1.12262	2.22104	0.56459	1.09202	2.19943
	5	0.61366	1.2748	2.5501	0.62496	1.2509	2.49379	0.63225	1.26157	2.50682	0.63836	1.25774	2.47685
4000	2	0.72241	1.39973	2.83001	0.69922	1.36145	2.66832	0.67913	1.32105	2.53718	0.67751	1.30977	2.62299
	3	0.89938	1.76487	3.58951	0.85358	1.73223	3.37386	0.89709	1.67892	3.24119	0.85957	1.68032	3.22716
	4	1.03653	2.10116	3.98855	0.9715	1.98775	3.83469	0.96385	1.87807	3.62008	0.95753	1.8564	3.68269
	5	1.25635	2.5306	4.80257	1.19211	2.54138	4.7413	1.19453	2.33547	4.48676	1.24731	2.32187	4.54973
5000	2	1.09231	2.15912	4.19843	1.07995	2.044	3.99808	1.06102	1.99967	4.05524	1.05971	2.02657	3.9953
	3	1.39378	2.70822	5.29489	1.37967	2.67659	5.11438	1.36784	2.66511	5.28403	1.35193	2.64737	5.23518
	4	1.41373	2.6871	5.32409	1.30994	2.65433	5.15203	1.33458	2.58562	5.16531	1.31004	2.57086	5.17844
	5	1.77974	3.53069	6.93992	1.72774	3.38791	6.72808	1.63398	3.32734	6.50819	1.61203	3.24615	6.37525