Coordination Models for 5G Multi-provider Service Composition: Specification and Assessment

George Darzanos, Manos Dramitinos, and George D. Stamoulis

Athens University of Economics and Business (AUEB)
Athens, Greece
{ntarzanos,mdramit,gstamoul}@ aueb.gr

Abstract. The inherently multi-stakeholder value chain of 5G services calls for business and service coordination. In this paper, we introduce and evaluate coordination models for the multi-provider service composition, namely the Fully Centralized, Distributed and per-Provider Centralized models, in the context of the 5GEx multi-provider orchestration framework. We perform a scalability assessment of the models in terms of the message overhead, also investigating the trade-off between service composition efficiency and message overhead. Our sensitivity analysis on the different parameters of our evaluation framework reveal that hybrid models scale better, but also other models may achieve the same level of message overhead under certain conditions.

1 Introduction

5G envisions services with new capabilities over a unified networking and cloud infrastructure impacting verticals such as Infotainment, e-Health, Energy, Automotive, Manufacturing Factories of the Future [1]. These services rely on an all-IP fully *softwarized* network architecture from core to edge that utilizes virtualized resources in order to orchestrate, trade, deploy and manage services jointly over the network, storage and compute domains in a fast, agile and secure, way. The 5G customer-facing retail services rely on wholesale infrastructure services, which can be categorized to *Connectivity*, *Virtual Network Function as a Service* (VNFaaS - network and application functions chained to support the service) and *Slice as a Service* (SlaaS - a managed set of Connectivity and VNFaaS services, additionally providing to the customer full control and management access) [2].

The value chain of 5G services inherently involves multiple stakeholders and administrative domains, each contributing to the end-to-end service provisioning. Network Service Providers (NSPs), Network Function Providers, Infrastructure Service Providers (IfSPs), Over-the-top Providers, are only a subset of the stakeholders being part of the 5G ecosystem. This greatly complicates the task of end-to-end service composition and inter-provider coordination, thus the adoption of sophisticated service Orchestrators is vital. The way Orchestrators are organized, how and what information is exchanged amongst them has great impact on the efficiency of the 5G service composition. In this paper we specify concrete coordination models for service composition in the 5G multi-provider setting,

which are generic enough to apply to any underlying 5G orchestration framework. We perform qualitative and quantitative, simulations-based assessment of them. We assess their scalability in terms of message overhead and service availability, providing recommendations regarding the information dissemination and management policies over the 5G architecture and service model.

2 5G Exchange Framework

5GEx [3] is an open multi-service multi-operator inter-networking approach for orchestrating, trading and composing 5G infrastructure *wholesale services*. Through the 5GEx framework NSPs and Clouds trade, orchestrate and manage services on the fly, so as to meet end user demand for 5G retail services. The fact that there are multiple ways to do this, motivates the work reported in this paper regarding coordination models for service composition in 5G.

The 5GEx architecture, depicted as Fig. 1, anticipates and specifies standard interfaces, extending the ETSI MANO architecture to the multi-provider setting of 5G services. A Multi-provider Multi-domain Orchestrator (MdO) orchestrates services over multiple technology and administrative domains using multiple Domain Orchestrators. The MdO interacts with Domain Orchestrators via Interface (3) to orchestrate resources and services within the same administrative domain and interacts with other MdOs over Interface (2) to request and orchestrate services across domains. The MdO exposes over Interface (1) service specification APIs that enable the Enterprise Customer, i.e. an Online or Network-Cloud Service Provider to demand a service. 5GEx also considers third party providers, which do not own resource domains but operate MdO to broker resources and services from other providers.

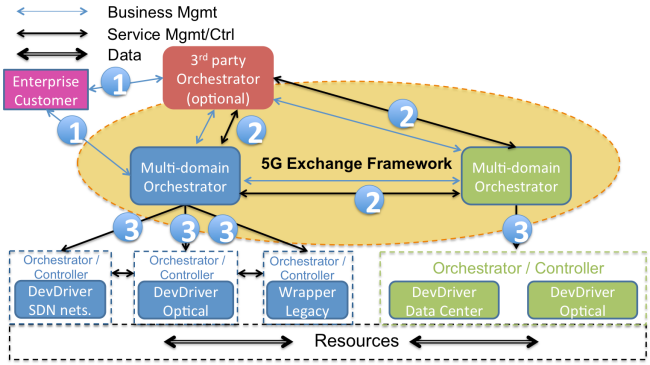


Fig. 1: Simplified conceptual architecture of 5GEx.

3 Related Work

To the best of our knowledge our work is the first study of coordination models for 5G multi-provider service composition. However, there is some related work

in other contexts, such as DiffServ, Brokers, Grids and Web services. Regarding Brokers, the necessity for coordination models to manage multi-agent systems led to agent-oriented coordination models and collaboration patterns [4]. Similarly, a bandwidth broker architecture for scalable end-to-end network services of guaranteed quality is introduced in [5] and [6]. These broker architectures relate to 5G Orchestrators, dealing with QoS management and admission control of multi-domain network services. However, contrary to our paper, these works focus only to the network domain, ignoring compute/storage aspects, also lacking an exhaustive investigation of the alternative hierarchies and their properties. Closely related to the hybrid approach of our paper is [7], where a virtual and dynamic hierarchical architecture for a scalable e-Science Grid is introduced, based on the notion of virtual groups, consisting of grid nodes that are within the same domain, have similar properties and exchange information frequently. In particular, a three-layers hierarchy of virtual groups is proposed for scalable node discovery and service provisioning. One node with each group act as a coordinator and it is responsible for the information propagation toward all other groups. This is similar to our hybrid hierarchical approach with multiple Orchestrators. However, contrary to the coordinator nodes that only acts as relays, our 5G Orchestrators performs information aggregation, bundling and filtering. Finally, regarding Web services, an Internet-scale model for servers-to-clients asynchronous event dissemination is specified in [8]. After exploring the design space of a proxy-based architecture, a hierarchy of event forwarding proxies to deliver events from each source to each related receiver is proposed. Again, our 5G Orchestrators are more intelligent and have more functionalities compared to the forwarding proxies that only reduce the extent of the redundant information.

4 Coordination Models for 5G Service Composition

4.1 Specification Methodology

Prior to presenting the coordination models, we specify the solution space and a baseline scenario and illustrations so as to facilitate the reader. The main design aspects of coordination models for 5G service orchestration are:

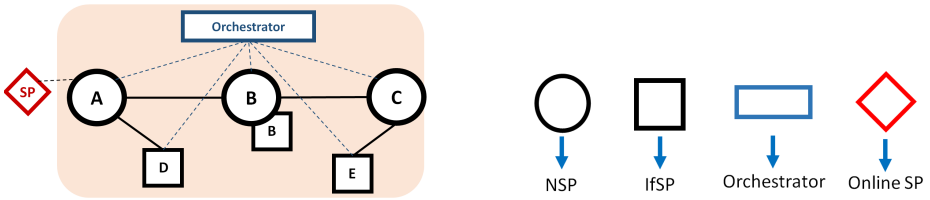
(i) Distributed vs Centralized: Service exchange and trading may be done in a fully distributed fashion through bilateral (possibly cascading) communications, or by means of a central entity, namely an Orchestrator that serves as the focal point for the aggregation/dissemination of information and service orchestration.

(ii) Fully Centralized vs per-Provider Centralized: Centralized models may be Fully or per-Provider Centralized. In the Fully Centralized model, a single Orchestrator does the orchestration for all 5G providers. In the per-Provider Centralized model multiple Orchestrators of multiple providers co-exist, each serving a different cluster of 5G providers. The providers of each cluster communicate with their Orchestrator according to the Fully Centralized model, while the Orchestrators of different clusters communicate in a distributed way. However, contrary to the Distributed model, each Orchestrator can contact all other Orchestrators regardless whether they are directly connected or not.

(iii) Coordination model phases. Every coordination model inherently consists of two phases, namely the *publishing phase* and the *service composition phase*. The *publishing phase* specifies the extent and granularity of the information exchanged among the providers regarding the service offerings supported. The publishing phase precedes the *service composition phase* that is triggered when a customer request arrives at a provider who uses the information that has been revealed in the publishing phase to compose the service.

(iv) Push vs Pull: The major difference of the pull to push models is the extent and type of information exposed at the publishing phase. In particular, in the push model the providers publish SLA offers, i.e. full service specifications prior to any customer request. On the other hand, in the pull models, each provider's service capabilities are published, a generic aggregate-level set of service types, QoS attributes and price ranges. An actual SLA offer is generated only after a customer's request.

The aforementioned options result in eight generic coordination models, defined below. For the better presentation of the coordination models we consider a specific scenario, depicted in Fig. 2a, with multiple providers operating under an orchestration framework such as 5GEx. The common support of the orchestration framework is depicted by the colored rectangle enclosing the providers. In our scenario, A and C are NSPs, D and E are IfSPs of compute and storage, and B is both NSP and IfSP. SP is an On-line Service Provider who needs a multi-provider service. SP has already an established business relationship with at least one NSP, e.g. in order to purchase connectivity. We henceforth refer to this provider (i.e. A in our scenario) throughout the paper as the primary provider for SP. Note that in the Centralized models, only the providers of the orchestration framework are aware of the Orchestrator's existence and not S, thus SP **always** contacts his primary provider. Fig. 2b introduces some basic notation and illustrations that will be used throughout the paper.



(a) The 5G ecosystem of our scenario.

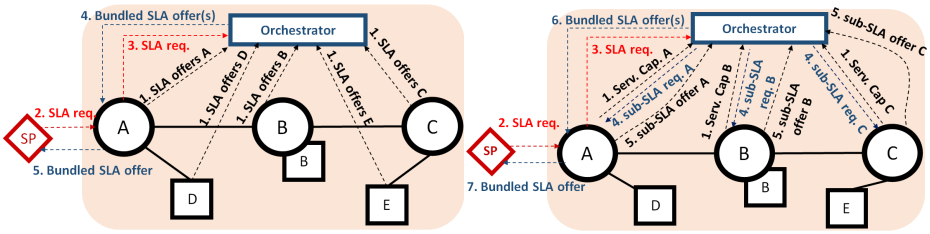
(b) Basic notation and illustrations.

Fig. 2: The 5G ecosystem and actors of our scenario.

4.2 Fully Centralized Models

Push Model. During the *publishing phase* all providers submit to the Orchestrator their service offers, i.e. their Service Catalogue entries in the form of SLAs (step 1 of Fig. 3a), which contain on-net destination(s), QoS attributes, price and offer expiration time. The Orchestrator, uses the topology view and the providers' offers gathered during the publishing phase to perform centrally the

service composition phase for each customer request passed to him (step 3) by some provider that receives it (step 2): The Orchestrator computes a bundle of SLA offers meeting the request and returns a solution to the primary provider (step 4), which in turn returns it to the customer (step 5).



(a) Fully Centralized push. (b) Fully Centralized pull.

Fig. 3: Exchange of messages for the Fully Centralized models.

Pull Model. Fig. 3b depicts the sequence of steps for the Fully Centralized pull model. In step 1, the providers publish to the Orchestrator their service capabilities. Again, the service composition phase is initiated upon a customer's request arrival to the primary provider (step 2). The Orchestrator uses the service capabilities collected during the publishing phase to send (sub-)SLA requests to the providers able to satisfy (part of) the request (step 4). For instance, the Orchestrator may push a sub-SLA request only for compute and storage resources to D. Then, these providers reply with offers (step 5) to the Orchestrator, which consolidates them and pushes one or more bundled SLA offers to the primary provider (step 6). Note that, in Fig. 3b, we only depict the steps for the subset of providers that the Orchestrator determined as highly possible actors of the current service chain (A-B-C). However, the publishing phase precedes and does not depend on service requests, thus step 1 applies to all providers. We use this simplification for all the pull models presented in this paper.

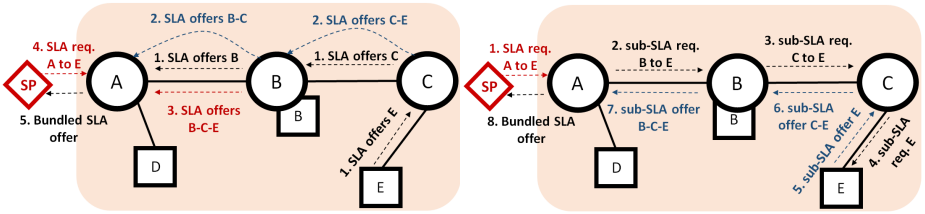
4.3 Distributed Models

Distributed models rely on bilateral cascading of service capabilities or SLA offers. This means that each provider communicates only with his direct neighbors.

Push Model. During the publishing phase each provider exchanges SLA offers with all of his direct neighbors. Each provider can also *bundle* his own SLA offers with those received, and then advertise bundles to his other neighbors.Through the bundling process, a provider can gradually increase the distance (in hops) that his bundled SLA offers can reach, as described in the next paragraph.

Fig. 4a depicts the exchange of messages for the Distributed push model. For demonstration purposes, we do not present all the exchanged messages, but we focus in a specific chain (A-B-C-E) in order to show how an offer from A to E is created by means of bundling. In the first iteration (step 1), the providers exchange only their own offers, thus only offers of maximum hop count of two can

be created. In each step, the providers use the information gathered in previous steps to create the bundled offers and increase the hops. Thus, after the third iteration (step 3), provider A has received an SLA offer from B that enables him to build a chain to provider E. Whenever SP requests a service from A to E (step 4), service composition is triggered and A can respond immediately because of the bundling done during the publishing phase.



(a) Distributed push. (b) Distributed pull.

Fig. 4: Exchange of messages for the Distributed push models.

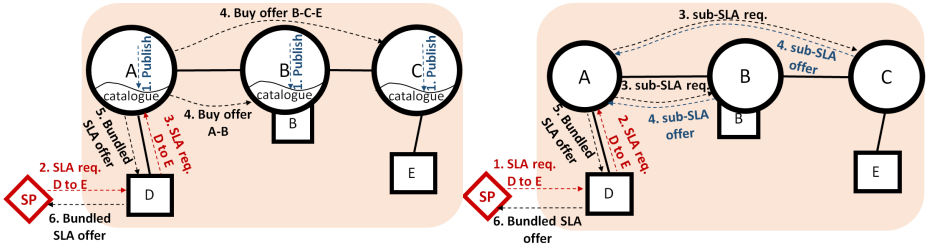
Bundling only applies to network services, while compute and storage SLA offers are forwarded as received. Bundling or forwarding all the SLA offers coming from the neighbors may create flood as the length of the service chain becomes large. This motivates smart information dissemination policies taking advantage of the topology hierarchy to avoid flooding, e.g. by defining a maximum length of the bundled SLA offer path. On the other hand, too conservative bundling policies though may lead to low offer availability of multiple hops offers.

Pull Model. In the publishing phase of the Distributed pull model, the bundling process we described in the previous paragraph is performed on the announced service capabilities. Contrary to push model, the providers exchange messages also during the service composition phase, as depicted in Fig. 4b. Once the primary provider receives a request (step 1), he extracts the part of the SLA that he cannot satisfy himself. Then, he uses the service capabilities collected at the publishing phase to determine his neighbors that can satisfy the remaining part of the SLA and sends the respective sub-SLA requests (step 2). Each provider receiving a sub-SLA request applies the same process until the request reaches the destination (step 3). All providers receiving a request return a sub-SLA offer in the reverse order of requests until the bundled offer reaches the primary provider (step 7) that delivers the final offer to the customer (step 8).

4.4 Per-Provider Centralized Models

Push Model. The publishing phase is performed in each cluster separately and it is followed the same process as in the Fully Centralized push model (Fig. 3a). Thus, each Orchestrator (A, B, C) acquires full knowledge for the SLA offers within its cluster. As presented in Fig. 5a, these offers are published to a Service Catalogue that is accessible by any other Orchestrator (step 1). Again, the service composition phase is initiated by a customer request (step 2). After

receiving the request, the primary provider D forwards the request to the local Orchestrator A (step 3). Then, A calculates the path to the destination and browses the Service Catalogues of the Orchestrators that are part of the service chain. After the evaluation of the available offers, A purchases the desired sub-SLA offers and bundles them himself (step 4). Following the same logic as in the Fully Centralized model, the Orchestrator A returns a bundled offer to the primary provider (step 5), which in turn returns it to the customer (step 6).



(a) Per-Provider Centralized push. (b) Per-Provider Centralized pull.

Fig. 5: Exchange of messages for the per-Provider Centralized models.

Pull Model. During the publishing phase of the pull, service capabilities are exchanged within each cluster and stored to the Orchestrators' Catalogues. As shown in Fig 5b, the service composition phase is initiated by a customer request (step 1). The local Orchestrator A, computes the service chain based on the service capabilities and sends sub-SLA requests (step 3) to the other Orchestrators involved in the service chain, bundles the received (step 4) sub-SLA offers and delegates the bundled one to the primary provider (step 5). Finally, the primary provider returns this offer to the customer (step 6).

5 Assessment

In this section, we perform a scalability assessment of the proposed models based on total number of messages exchanged among the service orchestration actors. We investigate how the scalability of the proposed models is affected by different parameters of the ecosystem. Also, we examine aspects such as SLA offers availability and redundancy of exchanged messages.

5.1 Methodology

Topology. We simulate an environment of multiple Transit-NSPs (T-NSPs), Edge-NSPs (E-NSPs) and IfSPs being interconnected in a hierarchical topology of three tiers, resembling the Internet tiered hierarchy: The first tier contains all the T-NSPs connected in a full-mesh fashion, each of them serving a number of E-NSPs from the second tier. With probability 0.5 an E-NSP has a peering link with another randomly selected E-NSP. The IfSPs of the third tier are uniformly distributed connected to the E-NSPs, while with probability 0.5 an IfSP is connected to two E-NSPs. In the Fully Centralized models we assume

that the Orchestrator joins the full mesh of the top tier. This means that if an IfSP sends a message to the Orchestrator, it will cross the AS of three different providers in the physical topology resulting in a count of 3. In the per-Provider Centralized models we assume that only the T-NSPs maintain an Orchestrator, therefore the number of clusters created equals the number of T-NSPs.

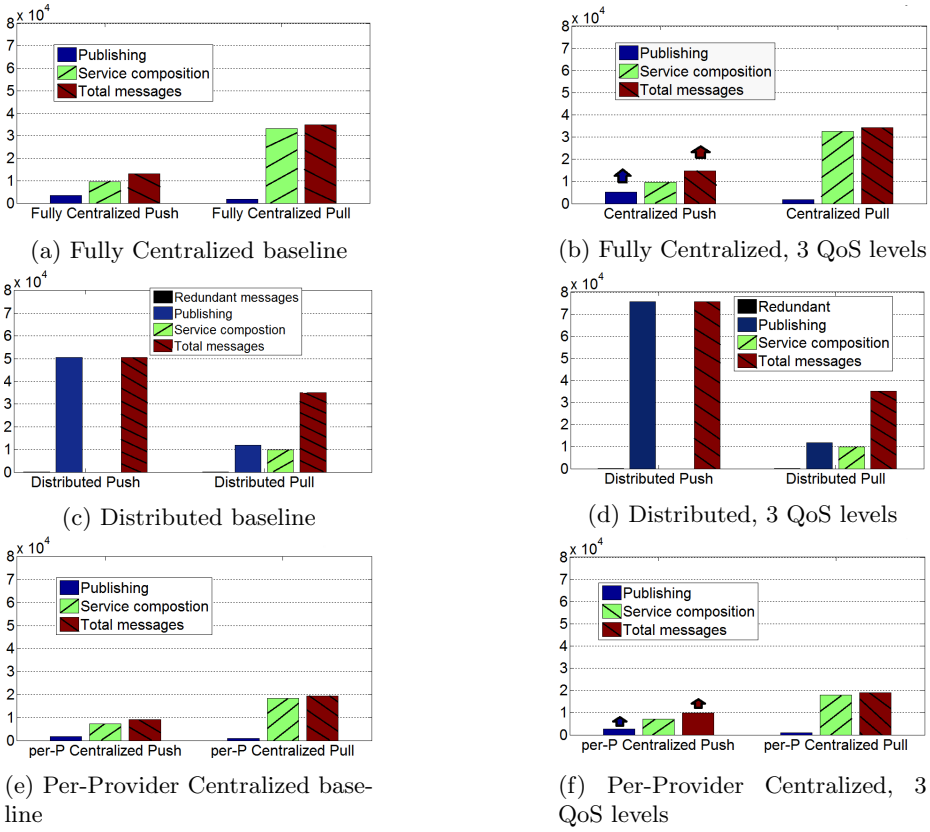
SLA offers and Service capabilities. We categorize the services that a provider can offer to network (N), compute (C) and storage (S) domain services. We assume that the T-NSPs offer services only in N domains, E-NSPs in all domains, while IfSP in C and S domains. We assume that each provider offers various service types in each domain. In the push models, a provider may create SLA offers of multiple QoS levels for the same service type; therefore, the total number of SLA offers is also depends on the number of different QoS levels. In the pull models, each provider creates only one service capability for each service type because service capabilities are more generic compared to SLA offers, thus can be more compacted. In the Distributed models we investigate different levels of bundling intensity, i.e. different thresholds on the maximum length of the bundled SLA offer (or service capability) path. Finally, for the forwarding of C/S SLA offers and service capabilities the providers takes advantage of topology hierarchy to reduce the number of duplicates.

Service requests. We assume that the service requests are generated at the edge, hence received at E-NSPs and IfSPs. The requests coming from the customers of an E-NSP demand connectivity from their primary E-NSP to a remote PoP or IfSP, but may also request compute and storage to the source or destination provider. The requests arriving to an IfSP can be of the same type with that of E-NSP customers, or it can be a request for C or S resources in multiple IfSPs with optional connectivity between them.

5.2 Scalability Assessment and Sensitivity Analysis

We ran multiple experiments over different topologies generated as described in the previous subsection. We use the results of a single simulation setup as baseline to compare the different model's performance, and then we perform a sensitivity analysis on the ecosystems parameters. Our baseline simulation setup parameters are set to: T-NSP=5, E-NSP=20, IfSP=40, 5 PoPs per E-NSP, 1 PoI per neighbor, 2 levels of QoS per service type, 1 service capability per PoI pair, 30 service request per E-NSP and IfSP. In the Distributed models, we assume that the providers perform *intense bundling*, thus they can reach any destination within the 5G orchestration framework.

Single-setup observations on message overhead. Fig. 6 depicts the message overhead of all models under two different setups. The first one is the baseline setup, while the second one has 3 levels of QoS per service type, i.e. one more compared to the baseline. Focusing on the baseline setup, we can observe that the higher message overhead is observed in the Distributed models, while the per-Provider Centralized models are the ones that generate the fewest messages. As expected, the pull models generate fewer messages than push during



(a) Fully Centralized baseline

(b) Fully Centralized, 3 QoS levels

(c) Distributed baseline

(d) Distributed, 3 QoS levels

(e) Per-Provider Centralized baseline

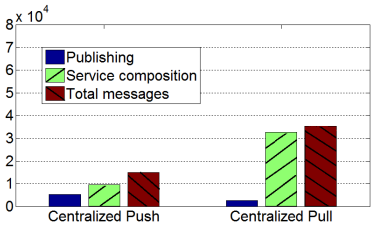
(f) Per-Provider Centralized, 3 QoS levels

Fig. 6: Message overhead for all coordination models under two different setups.

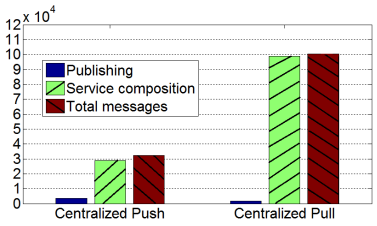
the publishing phase, since the service capabilities are more compacted than SLA offers. The push models have an advantage in the composition phase since they require the exchange of fewer messages for the composition of each service. Finally, the duplicates created because of the bundling and forwarding actions are negligible since providers take into account the overall topology for message propagation.

Impact of the number of available QoS levels. The number of available levels of QoS per service type does not affect the pull models since 1 service capability message that covers all QoS levels will be pushed. On the other hand, the push models are affected since a different SLA offer will be pushed for each QoS level (Fig. 6b, Fig. 6d, Fig. 6f). We can observe that Distributed push is the most "sensitive" in the number of QoS levels and SLA offers, due to the intense bundling/forwarding. The Fully and per-Provider Centralized push models are also affected, but they are less sensitive.

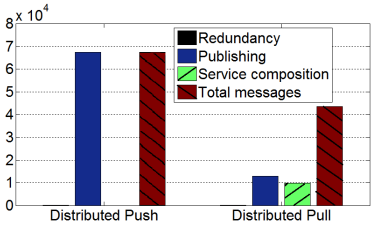
Impact of the number of Edge PoPs. Fig. 7a, Fig. 7c and Fig. 7e depict the message overhead for all coordination models for a setup with the double number



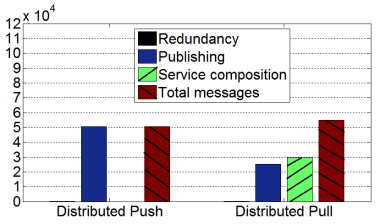
(a) Fully Centralized, 10 PoPs per E-NSP



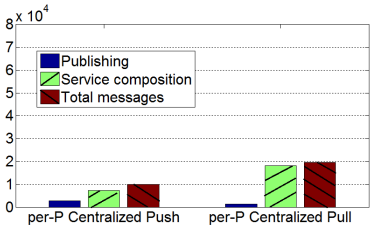
(b) Fully Centralized, 90 requests per E-NSP and IfSP



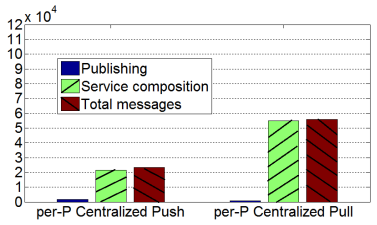
(c) Distributed, 10 PoPs per E-NSP



(d) Distributed, 90 requests per E-NSP and IfSP



(e) Per-Provider Centralized, 10 PoPs per E-NSP



(f) Per-Provider Centralized, 90 requests per per E-NSP and IfSP

Fig. 7: Message overhead for all coordination models under two different setups.

of edge PoPs per E-NSP compared to the baseline. Again, the Distributed models are affected more than the Centralized ones where the impact is minor. The number of edge PoPs affects the message overhead of push and pull models in the same extent, as depicted in Fig. 6c and Fig. 7c.

Impact of the number of service requests. Fig. 7b, Fig. 7d and Fig. 7f depict the message overhead of all coordination models for a setup with three times more service requests compared to the baseline setup. The number of requests affects the service composition phase of each model, with the Centralized pull affected the most since the Orchestrator must exchange increased number of messages with E-NSPs and IfSPs being at the edge of the network. The per-Provider Centralized pull is also affected, but not as much since the multiple Orchestrators are closer to the edge providers of their cluster. Interestingly, the Distributed pull model generates almost the same number of messages with the per-Provider Centralized pull.

Impact of the number of T-NSPs, E-NSPs and IfSPs. Table 1 shows the total number of messages exchanged for different topology sizes. The results reveal that increasing the number of IfSPs significantly affects the message exchange of all models, namely 85% increase in Fully Centralized, 180% in Distributed and 83% in per-Provider Centralized models. On the other hand, after an increase of the number of E-NSPs the message overhead is increased by 27% in Fully Centralized, 75% in Distributed and 16% in per-Provider Centralized models. Finally, the impact on a possible increase on the number of T-NSPs is even lower. We also observe that by doubling the total number of providers in the system, the message overhead is doubled in Fully and per-Provider Centralized models, but the increase is exponential in the Distributed ones.

Table 1: Message overhead for different topology sizes

			Push			**Pull**		
T-NSPs	**E-NSPs**	**IfSPs**	Fully Centr.	Distr.	per-Prov.	Fully Centr.	Distr.	per-Prov.
5	**20**	**40**	13022	50476	9026	34813	35016	19288
10	**20**	**40**	13286	54720	9226	34781	37026	19354
5	**40**	**40**	16414	87421	10887	43225	57301	23268
5	**20**	**80**	23952	138680	16673	59914	85496	33544
10	**40**	**80**	26932	207846	18651	70720	123536	39120

Bundling intensity and SLA offers availability. The aforementioned results reveal that the Distributed models do not scale due to the intense bundling and forwarding of the SLA offers and service capabilities. Thus, we investigate how a restriction on the maximum hops a bundled SLA offer can reach may mitigate this issue. We also examine how such restrictions may lead to low availability of SLA offers, hence customer requests for remote PoPs cannot be immediately satisfied. The results show that if the providers adopt a bundling policy of maximum two SLA offers, the message overhead is lower than all the other coordination models but the SLA offers availability drops to 19%. A bundling policy of maximum three SLA offers leads to an availability of 56% but for double the message overhead of the Centralized models. Note that without bundling an SLA offer path has length 2, while after bundling 4 SLA offers all destinations in our topology can be reached.

5.3 Discussion

Distributed models do not scale since they are highly affected by multiple parameters of the 5G ecosystem, including service types and QoS levels. Second, as the number of PoPs per E-NSP increases so does the number of possible destinations in the 5G ecosystem, thus the message overhead increases both for pull and push models. Finally, the message overhead increases exponentially with the total number of providers in the ecosystem.

Pull models are advantageous in the publishing phase due to the more compacted nature of service capabilities compared to SLA offers. Push models have

an advantage in service composition since they exchange fewer messages per service. Thus, pull models are more suitable for limited demand and early service markets, while the push models are best for mature, liquid markets.

Per-Provider Centralized models scales better than all the others. While the Fully Centralized models appear perform similarly with the per-Provider Centralized for small topology setups, as the number of providers and the service requests increases the performance difference becomes clearer. The advantage of per-Provider Centralized models lies on the fact that during the publishing phase the messages are pushed to closer distance (in hops) cluster-local Orchestrators.

6 Conclusions

In this paper we introduced multiple coordination models for 5G multi-provider service orchestration. We simulated an Internet-like environment of multiple 5G providers and evaluated the models under different setups, performing a sensitivity analysis on the different parameters of the ecosystem. Our results reveal that Distributed models scale significantly worse than Fully and per-Provider Centralized models. As the ecosystem becomes larger the hybrid per-Provider Centralized models scale best. Evaluating the coordination models over different topology structures and further assessing smart bundling policies for the Distributed models comprise directions of future work.

Acknowledgments. This work has been performed in the framework of the H2020-ICT-2014 project 5GEx, which is partially funded by the European Commission. This information reflects the consortiums view, but neither the consortium nor the European Commission are liable for any use that may be done of the information contained therein.

References

1. The 5G Infrastructure PPP whitepapers. `https://5g-ppp.eu/white-papers/`
2. "NGMN 5G White Paper" by NGMN Alliance, 2016. `https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf`
3. A. Sgambelluri et al.: Orchestration of Network Services Across Multiple Operators: The 5G Exchange Prototype. In EuCNC (2017). `http://www.5gex.eu/`
4. Hayden, S., Carrick, C., Yang, Q.: Architectural design patterns for multiagent coordination. In: Proceeding of the International Conference on Agent Systems. vol. 99, (1999)
5. Zhang, Z., et al.: Decoupling QoS control from core routers: A novel bandwidth broker architecture for scalable support of guaranteed services. In: ACM SIGCOMM Computer Communication Review (2000)
6. Duan, Z., et al.: A core stateless bandwidth broker architecture for scalable support of guaranteed services. IEEE TPDS 15.2, 167-182 (2004)
7. Lican, H., Wu Z., Pan Y..: Virtual and dynamic hierarchical architecture for E-science grid. International Journal of HPCA 17.3, 329-347 (2003)
8. Yu, H., Deborah E., Ramesh G.: A hierarchical proxy architecture for Internet-scale event services. In: Proceeding of IEEE 8th WET ICE (1999)