Towards a Context Adaptive ICN-based Service Centric Framework

Peyman TalebiFard†, Ravishankar Ravindran‡, Asit Chakraborti‡, Guoqiang Wang‡, Victor C.M. Leung†
†University of British Columbia, Vancouver, BC, Canada. {PeymanT, VLeung}@ece.ubc.ca
‡Huawei Research Center, Santa Clara, CA, USA. {ravi.ravindran, asit.chakraborti, gq.wang}@huawei.com

***Abstract*—Recent shift in the networking paradigm towards networking of information enables semantically rich interconnection and open interaction among the entities of well known services like video distribution and emerging applications like Internet of Things (IoT). Information-centric Networking (ICN) realizes a rich networking layer with ability to support computing and caching usable by any application. The success of ICN depends on enabling new services for which an ICN-based edge service framework was proposed. This paper extends our proposal by elaborating on service functions to enable contextual interaction between users and services. Towards this, an overview of the proposed ICN-based edge cloud framework is provided, with comprehensive discussion on service composition, orchestration, and routing logic with mapping to resources in the underlying substrate. In this work, we elaborate on an ICN perspective of a self managing network. We formulate the problem with optimization problems at different stages consisting of user's context and requirements, Service Level Agreement (SLA) and available resources. We demonstrate the feasibility of the proposed approach by means of simulation and analytical reasoning.**

I. INTRODUCTION

With a rapid increase in globalization and availability of information from heterogeneous sources, such as social networks, sensors, user devices and etc., the need for an intelligent context aware service composition is inevitable. Information-centric networking (ICN) paradigm surveyed in [13] enables context-aware personalized service composition. ICN is a shift in networking paradigm that allows applications, services and networks to interact using network primitives centered around the information to be transferred. ICN decouples applications from the transport layer by first naming entities such as applications, services, and content and then binding consumers to them through a scalable name resolution infrastructure. This decoupling also assists with dynamic features like mobility, migration, and replication of named resources such as devices, content, or services. With such a shift in networking paradigm from traditional client-server Transmission Control Protocol/Internet Protocol (TCP/IP)-based towards networking of information, semantics of data can provide information pertaining the context of a user, service or application which yields a more intelligent service composition, better utilization of underlying infrastructure and tackling the mobility and dynamics of network in time and space dimensions. Furthermore, this may lead to the reduction of Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) for service providers. As an example, CCNx [1] is an instance of the ICN implementation. In this implementation, data location dependency is eliminated which allows reduction of resource consumption and makes the network utilization more efficient through caching and lead to CAPEX reduction. Furthermore, OPEX can be reduced by a more efficient caching methodologies as well as facilitating management and configuration such as path optimization at the strategy layer.

While the composed services are agnostic of the underlying infrastructure, they do require the compute, storage, networking resources to support certain capabilities, e.g., a certain level of Service Level Agreement (SLA). Composition of services can leverage contextual information such as location, preferences, device capabilities and social connections collected from users' profiles, devices, and access network infrastructure to deliver an optimized composition that meets the SLA, user/application requirements and resource constraints. Realization of network functions on generic platforms is an opportunity to realize ICN-based services in future networks. Towards this, we proposed [9] an incremental and generic ICN-based platform that supports ICN applications as edge-cloud services ranging from enterprise applications, content distribution, and services pertaining to the advancement of applications in Machine-to-Machine (M2M) and Internet of Things (IoT). In this paper a configurable and context-aware service orchestration platform is presented and based on this platform we propose a service composition methodology. Such a platform will manifest the idea of pushing the frontier of computing services and applications away from centralized nodes to the logical extremes of the network. These edge-cloud realizations can be located in the vicinity of the operator's Central Office (CO) or at Points-of-Presence (PoP), enabling local instantiation of ICN services to consumers, while providing global service delivery through a unified service control infrastructure.

II. RELATED WORK

Recent literature questions the motivation for ICN [4], and suggests incremental Hypertext Transfer Protocol (HTTP) to achieve ICN features. The comparison is not justified as ICN is envisioned as a future network protocol to work in both infrastructure and ad-hoc situations to adapt to extreme levels of dynamism, and to serve applications ranging from content distribution to IoT applications such as local device-to-device or Vehicle-to-Vehicle (V2V) applications. ICN enables

QSHINE 2014, August 18-20, RHODES, Greece

DOI 10.4108/icst.qshine.2014.256426

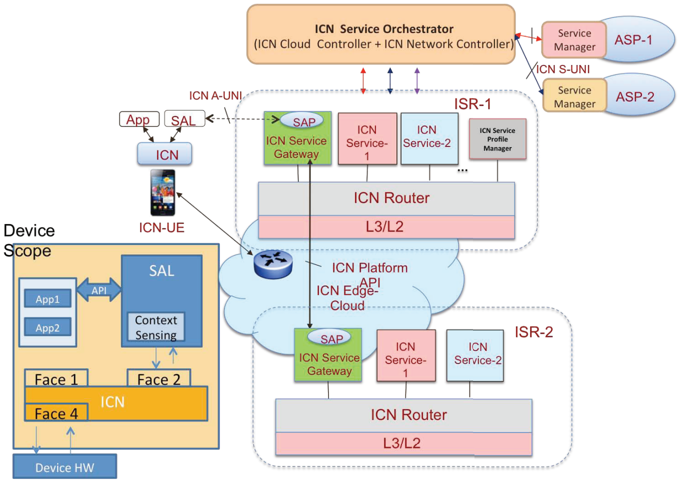


Fig. 1. ICN-based service centric edge-cloud framework.

these through features such as location independent name based routing, multicast/anycast, in-network caching, mobility support, and packet level security. In contrast, HTTP was primarily designed as an application layer protocol over a reliable host-to-host connection-oriented logic. To support service-centric networks and to improve flexibility, several protocols such as SERVAL [6] and openADN [7] have been proposed to enable efficient service/application delivery. While SERVAL is not an ICN, it proposes a name based service layer for multi-path service access and routing; [7] leverages SDN to enable application Protocol Data Units (PDUs) to be encapsulated by application-centric labels over which service delivery logic is applied by an openADN aware infrastructure. Other desirable protocol features have been proposed as extensions to handle content delivery as in the S-GET function in HTTP [8] or LISP [3] to handle the feature of ID and Location separation. All these protocol features form a subset of ICN capability, as it is a culmination of several future Internet architecture proposals studied over many years.

Current service composition approaches are mainly focused on reducing the failure rate of composed web services and maintaining a certain level of availability and reliability. Composition of services however, can not accommodate heterogeneity of service domain with different set of specifications and descriptions and does not consider the access network infrastructure and resources. ICN in the context of IoT/M2M enables a semantic interpretation and a semantic based interaction among heterogeneous entities of the whole system whereby the context of end users are considered and resources are allocated based on the SLA and user requirements. Furthermore, semantically capable ICN-based services can benefit from models that represent a higher level view of objects interactions [12]. This paper extends our previous work [9] that proposed a service-centric framework over ICN where the emphasis was on service contextualization and customization. This paper focuses on various aspects of service contextualization and service composition to handle user and application generated events.

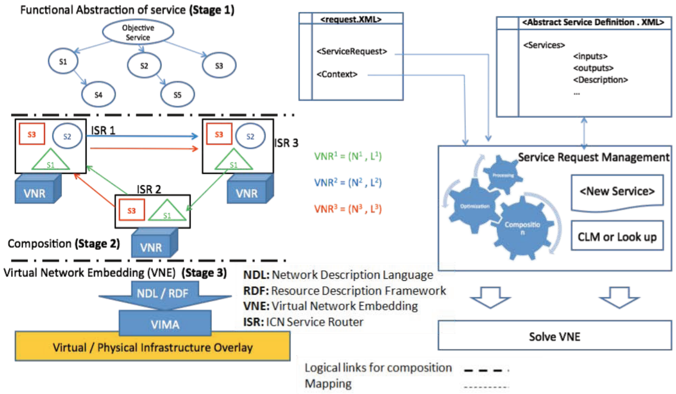


Fig. 2. Stages of service composition from abstraction to infrastructure substrate.

III. SERVICE CENTRIC EDGE-CLOUD FRAMEWORK

An ICN-based service centric edge cloud framework should be capable of leveraging the contextual interaction of entities such as users, services and network infrastructure to dynamically adapt to changes and minimize service interruption and SLA violations. The platform is aimed for ICN-based applications such as conferencing or IoT applications that can leverage name based content dissemination, in-network caching, receiver-oriented interest and data multicasting, content level integrity,and privacy. Such a system should support open-APIs, adapt to user dynamism through adaptation to changes of context, be capable of dynamic functional composition and enable different ICN protocols and related services for future deployments and experimentation. An end-to-end view of our framework is shown in Figure 1. In the spirit of incremental deployment, the proposed ICN platform is realized as ICN router hosting ICN services (we call this ICN Service Router (ISR)) overlaid over current IP network, with potentially multiple ISRs within the scope of an edge-network. All the ICN platform components can be realized as virtual functions; however for performance reasons the ICN router can be non-virtual as it performs ICN functions of name based routing and caching to all the service traffic passing through it. The important functional components as shown in Figure 1 are as follows. *ICN Service Platform*, *ICN Service API*, *ICN Service Orchestrator*, and *ICN cloud controller*. Detailed description of these modules are elaborated in [9].

IV. SERVICE ORCHESTRATION AT THE EDGE CLOUD

In this section we state the problem and the flow of procedure from service abstraction to service composition and finally realization of the composed service. Given the vector of contextual information for a specific user upon requesting a service, and the enforced SLA requirements, we show the work flow of the system from service abstraction to implementation. Figure 2 shows the high level procedure for the proposed approach.

Service-centric ICN paradigm manifest an object oriented networking paradigm where objects can be services, information objects or multimedia content objects. Service composition capability is a key design principle that applies within the Service Oriented Architecture (SOA) design principle that suggests the design of services in a manner that can be reused in multiple solutions that themselves may be made up of other composed services. The ability of a service to be recomposed should ideally be independent of the size and complexity of the service composition. Service composition in the context of this work is a more generalized vision that incorporates the ICN perspective which leverages the higher level view of object interactions.

In this work we make the following assumptions:

- Services can be looked up based on their service descriptions from a repository of existing services available as part of the ICN edge platform.
- Services are instantiated in a distributed (optimized) manner i.e in different locations in ISRs.
- Services are permitted to leverage each other's functionalities subject to access policies.
- We reduce the context vector dimensions to a limited number of attributes such as location, mobility, and device capabilities.

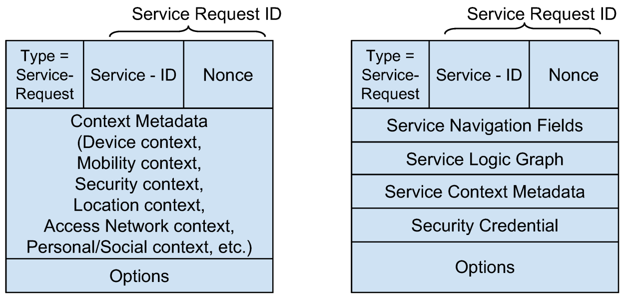
The ISR at the edge is in charge of discovery, composition and routing services over ICN. It enables features such as context-aware service composition and dynamic provisioning of services and applications for ASPs. The role of ICN-based edge cloud platform is to abstract the complexity and dynamism of the underlying network's infrastructure from the applications and service enablers. Two types of services can be realized, first type being the services that are accessed by consumers and the second type are auxiliary services used by ASPs (e.g. transcoding or data processing), also known as service enablers.

Contextual information can be managed at the core Service Access Point (SAP) and in part at the UE end by a *context management module* that is in charge of the following tasks [11]. *Context collection and aggregation* that involves collection of information from various sources (i.e. service context, user context or network context) and monitoring the dynamics of a service for context changes, *context processing* that involves modeling different types of context to perform filtering and transformation or inferring higher level context information for application specific interpretation, and *dispatch and distribution* of heterogeneous contextual information.

Context changes can be sensed by the UE and the network and such changes may trigger actions on both sides. This means that during a service session, running instance of a service can perform a dynamic functional composition on-the-fly due to state changes as a result of access point, device related, infrastructure related state changes or based on explicit expression by the UE.

Services can be pre-composed for different expected context for a given service, which can be mapped to the service request PDU. However, in some cases, given sufficient intelligence at the local service instance, and for cases not considered earlier, this can be done in real-time; the feasibility however depends on the complexity, performance requirement and practical considerations of the ability to handle it.

In the rest of this section we present the top down functionality of this reference architecture from service definition to service composition and infrastructure embedding. Figure 2 shows the high level work flow.



(a) Service Request PDU on ICN UNI-API (b) Service PDU on the network side

Fig. 3. Service request PDU on the UE and the network side

A. Service Abstraction and Identification

The service request can be represented in the form of an XML request which contains information about the user device, preferences, type of the request etc. Therefore, prior to the service abstraction and identification, processing of the service request is required. This stage involves the abstraction of components and links representing the connection among entities as well as description of services and required SLA. The data structure of a graph in the abstract representation is generic and can be reused across other domains and for other services. At the composition level, each of the abstract components will be translated into the required services and forming the service logic graph for the composition.

One possible to approach to infer the SLA based on the contextual information is forming hypothesis functions for different SLA parameters that can yield the SLA to some level of certainty. Given the context vector C as the set of features for decision making, and a set of parameters from a data set, the following hypothesis can be formed.

$$h(C) = \theta^T C$$

where θ^T is the transpose of an $n \times m$ matrix of m set of collected data for n features, and C is the context vector of size n.

B. Service Composition

The purpose of this phase as shown in Figure 2, is to form a service logic and build a graph of logical entities with logical links and mapping to virtual/physical entities (Compute, Storage, Network) for the VMs. A service logic graph identifies the sequence of services to be invoked. Service components can be aggregated and composed with sequential, parallel and selection operations.

The service graph generation is based on factors such as SLA requirements, network topology and congestion state, and load on the service instances. A service logic graph identifies the sequence of services to be invoked. It is rooted at the request-ID and connects through directed links to other nodes identifying service-ID respectively. This graph is interpreted at each ISR to determine the next-hop(s) to which service request has to be forwarded. A service logic graph does not need to be generated for each service request, rather a well defined set can be pre-orchestrated based on popularity of services being invoked, and incoming service interests arriving from users are mapped to it as they arrive over the A-UNI interface.

The composition scheme and the navigation vector is formed based on the service description specified through the XML representation. The composition engine as shown in Figure 2 extracts the service names and initiates interests with the PDUs shown in Figure 3 (b). In the PDU, the *service logic graph* field encodes the service execution logic and the *service navigation field* identifies the current service context, which when correlated to the service logic graph field, identifies the next set of services to be invoked. If the required service is locally available on the ISR, it would be identified by the forwarding engine of the ICN, otherwise the Forwarding Information Base (FIB), as in the CCNx implementation, will be queried to identify possible faces to propagate the interest. If the service is not found on the ISR and the FIB, an interest packet on the required service name will be sent through the all the faces and an entry will be added to the Pending Interest Table (PIT).

Once the composition scheme is determined and the navigation vector is formed, services are instantiated and substrates are formed as discussed in the next step. The interest packet towards a specific service for invoking, contains the context vector of the service consumer. Context changes are pushed from the Service Access Layer (SAL) on the UE side to add, scale out or scale up a specific service instance at the virtual substrate level or at the logical composition level. Examples of such changes include changing the device while a user using a personalized video streaming service.

The objective of service composition algorithms is primarily to find a feasible plan and then an optimal plan based on a set of optimization criteria. The purpose of this phase as shown in Figure 2, is to form service logic and build a graph of logical entities with logical links and mapping to logical entities (Compute, Storage, Network) for the VMs. The objective of this phase is form a service logic graph to meet the Service Level Objective (SLO) or the service to be composed.

Given the required minimum SLA parameters and the context vector C, we formulate the following maximization problem. Let $f()$ be the utility function of the composition that is dependent on context vector C, set of available finite set of services U and $SLA = (T, A, R)$ vector that consists of the attributes described earlier. To meet the SLA requirements, the essence of this stage is to form an SLA metric cube formed by independent SLA metrics of timeliness, availability, reliability and scalability. A cost objective function can be formed and minimized subject to the submitted SLA requirements by the consumer or as enforced by the service abstraction phase. The next step is forming the set of required Virtual Network Resources (VNRs) that is a set of required virtual nodes and virtual links.

C. From Service Logic Graph to Virtual Infrastructure Embedding

This phase involves the infrastructure and network substrate embedding that can be realized by network virtualization through embedding and instantiating virtual entities on substrate infrastructures that is broadly known as Virtual Network Embedding (VNE) [5]. Realization of a composed service involves the VM allocation methodology that can be done by forming virtual cluster of resources. In a Network Virtualization Environment (NVE), virtualized entities composed of network, compute and storage resources are deployed on a shared infrastructure. Therefore, the network virtualization problem involves allocation of virtual entities meeting the SLA requirements and minimizing the usage of resources. Resource allocation and optimization problem are addressed in the literature for embedding the virtual entities on shared infrastructure substrate [2]. VNE deals with the allocation of virtual resources both in nodes and links and can be divided into the sub problems of *virtual node mapping* and *virtual link mapping*. As described in details in [5], the VNE problem can be formally described as follows. Let the Substrate Network $SN = (N, L)$ be denoted by the substrate nodes N and the set of substrate links L. Furthermore, let $VNR^i = (N^i, L^i)$, $i = 1, \dots, n$ be a set of n virtual network requests where N^i and L^i represent the set of virtual nodes and virtual links respectively. Virtual nodes can constitute compute, storage or network resource nodes. Finally, for m number of resources globally, let $\dot{R} = \prod_{j=1}^{m} R_j$ be a vector space of resource vectors over the set of resources $R_1, ..., R_m$ and let $cap : N \cup L \mapsto \dot{R}$ be a function that assigns available resources to elements of the substrate network. For each VNR^i , we define a demand function $dem_i : N^i \cup L^i \mapsto \dot{R}$ as a function that assigns demands to elements of VNRs in terms of resources. A VNE problem can be formulated by the node mapping function f_i where $f_i : N^i \mapsto N \quad \forall n^i \in N^i : dem_i(n^i) \leq cap(f_i(n^i))$ and the link mapping function g_i where $g_i : L^i \mapsto SN' \subseteq SN, \quad \forall l^i \in L^i : \forall l \in g_i(l^i) : dem_i(l^i) \leq cap(l)$ for each VNR_i where g_i is a function that maps a virtual link of VNR_i to a link in Substrate Network (SN). Solving the above mentioned VNE problem is NP-hard [5]. Heuristic approaches should be developed to approximate the problem to a solvable approach.

For the purpose of evaluation and demonstration we formulate the above mentioned steps as a cost function optimization and use a greedy heuristic to obtain the best alternative. Given a set of services required to compose a new service on a set of ISRs located in different locations, we assume bandwidth and compute resources to be allocated for instantiation of the required services subject to meeting the required SLA parameters. Based on the nature of the newly composed service, some ISRs should be located at the edge and some ISRs may be best utilized near data centres. For example, to instantiate a video streaming service, the transcoding service would need to be near the data centre if the viewers of that content are geographically distributed. Therefore, these decisions are made at the abstraction and composition stages. To formulate the problem of link and node assignment, we start by defining the cost function as $Cost = \alpha \times bw(l^i) + \beta \times compute(n^i)$ for all $VNR^i = (N^i, L^i)$ over all services i where α and β are the cost coefficients. The following LP can be formed:

$$\begin{aligned} \text{Minimize} \quad & Cost = \alpha \times bw(l^i) + \beta \times compute(n^i) \\ \text{subject to} \quad & bw(l^i) \geq \sum_{i=1}^{M} f^{i,n}, \\ & \sum_{i=1}^{M} bw(l^i) \leq cap(L_i) \ \forall l^i \in L^i, \\ & \sum_{i=1}^{M} compute(n^i) \leq cap(N_i) \ \forall n^i \in N^i, \\ & f^{i,n} = \sum_{i} \lambda^{i,n} \ \forall i, n \in N^i. \end{aligned}$$

where $f^{i,n}$ is the flow of service i to ISR_n and $\lambda^{i,n}$ is the interest arrival rate of service i and M number of services. VM resource selection and allocation can be done by optimization algorithms. Except for the small instances of the problem where exact solutions can optimally solve the problem, solving larger instances of the problem is not trivial and demand heuristics techniques. However, heuristic methods may converge in a local optimum that is far away from the real optimum point. Metaheuristic methods tackle the issue of heuristic approaches by escaping from the local optimum and yield a near optimal solution. For the VNE problem we consider an embedding optimization problem with a cost objective that is a function of delay and compute resources and solve the problem with a Greedy algorithm approach.

V. EVALUATION

We evaluated the performance of the system presented above through modified version of the flow-level simulation tool Icarus [10] that is based on the Fast Network Simulation Setup (FNSS) toolchain. We chose the GEANT (European academic network) topology in the settings of the simulation tool and network routers have been assigned constant cache sizes. Requests have been modeled as Poisson processes while service popularity has been modeled with a Zipf distribution. We ran the simulation for 30000 seconds with a warmup duration of 1800 seconds. We considered a finite set of possible services (i.e. named as follows: video, audio, transcode, translate, charging and location) and each service can be chosen from 150 pre-defined candidates of the same name with different SLA parameters such as availability, delay and reliability. Each node is a virtualized element that can demand compute, network and storage resources and can have multiple ISRs as tenants. The inherent support of Icarus for caching and popularity of data chunks can partially mimic the underlying ICN behavior.

We evaluate the proposed method based on the network load that can impact maximum link utilization because it can greatly impact the performance of realtime applications and delay sensitive services such as conferencing, and multimedia streaming. The simulation results show the network load performance with varying the number of users and context change rate. Each context state is associated with specific SLA parameters. Figure 4 shows the maximum link load for settings with different number of users and varying context change rate or interest arrival rate for services. It can be observed that due to increase in the context rate change, link load and therefore link utilization is increased. In graph of Figure 5 we compare the case of not optimized VNE and the optimized VNE to demonstrate the scalability of the proposed methodology in response to the demand. For the case of non optimized VNE, the embedding is done solely based on the nearest resource load and available bandwidth and delay of possible links are not taken into account. Furthermore, in graph of Figure 6 we show that ICN performs better to the situations where same service items are requested that are more popular and this can manifest the impact of ICN caching feature on the improvement of the network utilization.

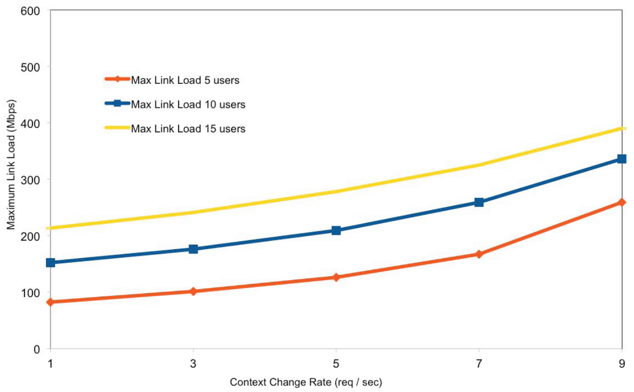


Fig. 4. Maximum link load for different number of users and varying context rate change and interest arrival.

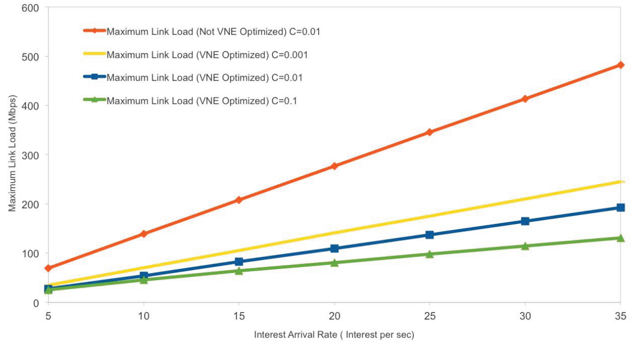


Fig. 5. Impact of VNE stage and scalability of the proposed metrology.

The problem of candidate service selection for the purpose of composition stage can be considered as a discrete optimization problem where the utility function expressed as a submodular function. Such problems can be solved by Greedy algorithms under the cardinality constraints and monotonicity by a guaranteed approximation factor of $(1 - \frac{1}{e})$. Therefore, a utility function $f : 2^U \mapsto \mathbb{R}$ can be defined that maps a subset of a ground set U to non-negative real numbers where f is monotone i.e. $\forall S \subset T \subseteq U : f(S) \leq f(T)$. Furthermore,

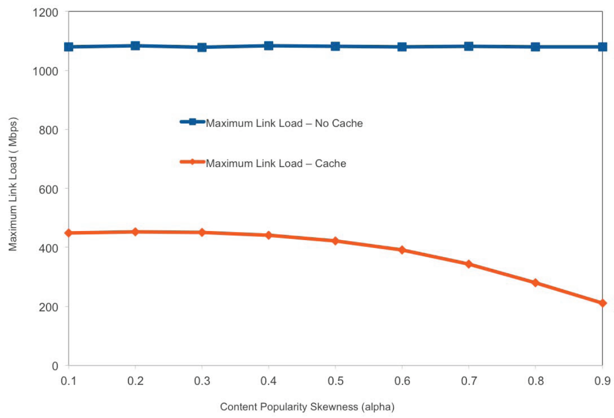


Fig. 6. Network load in response to services of different popularity.

f is a sub-modular function, i.e. if $\forall S \subset T \subseteq U, \quad \forall x \in U \setminus T : f(S+x) - f(S) \geq f(T+x) - f(T)$. In the following optimization problem, f is a utility function $f : 2^U \mapsto \mathbb{R}$ that maps a subset of a ground set U that represents the set of available services, to non-negative real numbers. f is monotone i.e. $\forall S \subset T \subseteq U : f(S) \leq f(T)$. Furthermore, we assume that f is a sub-modular function, i.e. if $\forall S \subset T \subseteq U, \ \ \forall x \in U \setminus T : f(S+x) - f(S) \geq f(T+x) - f(T)$.

$$\begin{aligned} \underset{\{S\subset U\}}{\text{Maximize}} \quad & f(S) = SLA(T,A,R) - Cost(bw, compute)\\ \text{subject to} \quad & sla_i \leq sla_i^{min},\ i = 1,\ldots,N.\\ & \sum_{i=1}^{M} r_i < r_i^{max}, \end{aligned}$$

where r is the available resources for M number of resources and $N = 3$ is the number of SLA parameters and T, A, R are delay, availability and reliability parameters. Solving the above mentioned problem however may be rather complex if solved as a global optimization problem. Considering the objective function f as a combination of $f_1(S_1)$ and $f_2(S_2)$ for SLA and infrastructure respectively, given that the choice of each element in the set of candidate services is dependent on the previous chosen element, a global optimization yields a problem size of $2^{(m+n)}$. Therefore, one possible approach is to decouple the service level objective function from the infrastructure resource allocation problem into two local optimization problems and reducing the size of the problem to the worst case of mn.

VI. CONCLUSION

ICN for the future cloud based Internet architecture enables context-aware personalized service composition. This leads to an object oriented networking approach where sensors, devices and users can participate towards a more efficient delivery and orchestration of services to meet the higher order needs of future users of communication systems. With a shift in networking paradigm from traditional client-server TCP/IP-based towards networking of information, semantics of data can provide information pertaining the context of a user, service or application which yields a more intelligent service composition, better utilization of underlying infrastructure and tackling the mobility and dynamics of network. ICN has been mostly seen in the context of content distribution, this paper elevates dynamic service interaction between consumers and services as the way to differentiate ICN from the current Internet. In this paper we discussed how the real vision of service-centric networking can be realized through virtualization at service, forwarding, and information plane realized through an the ICN network narrow waist. We proposed a novel design to realize ICN-based service orchestration at the edge cloud. We validated this by means of simulations and analytical reasoning and demonstrated the feasibility and scalability of the proposed methodology.

ACKNOWLEDGMENT

This work was supported in part by the NSERC DIVA Strategic Research Network, TELUS, Huawei and other industry partners.

REFERENCES

[1] Project CCNxTM. http://www.ccnx.org.
[2] N. M. K. Chowdhury, M. R. Rahman, and R. Boutaba. Virtual network embedding with coordinated node and link mapping. In *INFOCOM 2009, IEEE*, pages 783–791. IEEE, 2009.
[3] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis. The Locator/ID Separation Protocol (LISP). *draft-ietf-lisp (Experimental RFC)*, 2013.
[4] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker. Less pain, most of the gain: incrementally deployable ICN. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, SIGCOMM '13, pages 147–158, New York, NY, USA, 2013. ACM.
[5] A. Fischer, J. Botero, M. Beck, H. De Meer, and X. Hesselbach. Virtual network embedding: A survey. 2013.
[6] E. Nordstrom, D. Shue, P. Gopalan, R. Kiefer, M. Arye, S. Y. Ko, J. Rexford, and M. J. Freedman. Serval: An end-host stack for service-centric networking. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, 2012.
[7] S. Paul and R. Jain. Openadn: Mobile apps on global clouds using openflow and software defined networking. In *Globecomm*, December, 2012.
[8] L. Popa, A. Ghodsi, and I. Stoica. Http as the narrow waist of the future internet. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, pages 6:1–6:6, New York, NY, USA, 2010. ACM.
[9] R. Ravindran, X. Liu, A. Chakraborti, X. Zhang, and G. Wang. Towards software defined icn based edge-cloud services. In *Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on*, pages 227–235. IEEE, 2013.
[10] L. Saino, I. Psaras, and G. Pavlou. Icarus: a caching simulator for information centric networking (icn). In *Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques*, SIMUTOOLS '14, ICST, Brussels, Belgium, Belgium, 2014. ICST.
[11] P. Talebifard and V. Leung. A Dynamic Context-Aware Access Network Selection for Handover in Heterogeneous Network Environments. In *2011 Proceedings IEEE INFOCOM Workshops, April 10-15, 2011 in Shanghai, P.R. China.*, pages 385–390. IEEE, 2011.
[12] P. TalebiFard and V. C. Leung. Expansion properties of topology for networking of information in cloud. *IEEE Transactions on Parallel and Distributed Systems*, 99(PrePrints):1, 2013.
[13] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos. A survey of information-centric networking research. volume 16, pages 1024–1049, 2013.