An Information Centric Networking Approach Towards Contextualized Edge Service

Peyman TalebiFard†, Ravishankar Ravindran‡, Asit Chakraborti‡, Jianli Pan∗,
Anu Mercian§, 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
∗Department of Mathematics and Computer Science, University of Missouri - St. Louis, pan@umsl.edu
§School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ, amercian@asu.edu

***Abstract*—Information Centric Networking (ICN) has been a popular research topic in the last few years, but has not attracted industry attention because of its disruptive view; this is expected considering the evolution from PSTN to IP. Towards its adoption, ICN should not only address challenges raised by current applications, but also enable a compelling service framework for next generation of networking. We envision that in the next generation networks, the network narrow waist will allow an efficient distribution of intelligence across terminals, access, edge and core network. This will enable new applications, services and future business models to be realized. Two other technologies, NFV and SDN, which in essence are frameworks that enable service-centric networking, fit well with the objective of information-centric networking, where the delivered content is a result of contextual interaction between consumers and services orchestrated to meet service objectives. Most significant benefit of this interaction will be in the network-edge considering sensitivity to service latency, customization, and contextualization. This paper provides an overview of an ICN based edge service framework, with comprehensive discussion on service composition, orchestration, and routing logic with mapping to resources in the underlying substrate. We also provide a discussion of the prototype to realize this platform and a network based conferencing system scalable to large number of participants; however the platform itself is generic to handle any service type including content distribution, video conferencing, and M2M applications.**

I. Introduction

Information-centric Networking (ICN) surveyed in [17] is a shift in networking paradigm that allows applications, services and networks to interact using network primitives centred 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.

Two industry trends, Network Function Virtualization (NFV) and Software Defined Networking (SDN), enable agile service infrastructure and resource management for operators. NFV gained attention with the operators demanding [5] a platform to realize network functions in software over commodity hardware achieving economy of scale offered by cloud computing technology. SDN allows for independent evolution of the control and forwarding plane through network centric programmability complementing NFV's service virtualization objective.

Realization of network functions on generic platforms is an opportunity to realize ICN based services in future networks. Towards this, we proposed [12] an incremental and generic ICN based platform that supports ICN applications as network-edge 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) space. The platform leverages NFV constructs that allow the ICN framework to be realized as a *Virtual Network Functions* (VNF), these network functions are application functions in an ICN case. The platform enables service virutalization through open-APIs to service owners to automate provisioning and manage services (provision, scale, or migrate), while service flows are engineered by service specific controllers owned by the operator or the service owner. The platform enables an ICN-based user-to-network interface (UNI) that allows rich service level ICN interaction between consumers and the ICN platform which includes service features such as discovery, context expression, content or service request and response based on naming primitives. The context adaptation functions are realized through extensions of the ICN-APIs to accommodate variety of context parameters.

This paper discusses an ICN based edge service platform with focus on contextualized service delivery. 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-service 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. Following points motivates this realization:

Homogenized Application Delivery: Client-server model is the way to deliver services today, the assumption being that servers are fixed entities located in data centres to deliver services; this assumption no longer holds with NFV, as cloud computing can be enabled in an hierarchical manner as required by the service operator. Furthermore, to improve flexibility, several protocols such as SERVAL [9] and openADN [10] have been proposed to enable efficient service/application delivery. While [9] proposes a name based service layer for multi-path service access and routing; [10] leverages SDN to enable application PDUs to be encapsulated by application-centric labels over which service delivery logic is applied. Other desirable protocol features have been proposed as extensions to handle content delivery as in the S-GET function in HTTP [11] or LISP [6] to handle the feature of ID-Locator primitive. 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. ICN is a good candidate to homogenize the application delivery spanning network segments of heterogenous computing , storage, and bandwidth resources including sensor networks, Delay Tolerant Networks (DTN), Ad hoc environments, while complementing NFV and SDN objectives of service-centric networking in the infrastructure.

Service Customization and Contextualization: Services are best delivered by locally customizing them to what users want, because users who are located in different locations have different needs and requirements based on their context. Context can be defined as any information that is used to describe the state of an entity and can be classified as being user-, network-, service-, device- centric. Modeling and reasoning of heterogeneous contextual information involves a trade off between complexity of reasoning and expressiveness of data [14]. Mapping the contextual information into service level and network level requirements leads to the challenge of federated and standardized semantic representation of state and context. On the other hand, users themselves access services through heterogeneous devices, network connectivity, with subjective preferences. Moreover, mobility considerations require services to be delivered from the most vantage point. An edge service framework that can handle this dynamic requirements with minimum overhead is desirable. Context of users and services can be inferred and semantics (meaning of data items within a context ontology) of content predicates can be interpreted towards an intelligent dissemination of services and content items. In addition to a physical view of network topology, ICN based services and applications can benefit from a higher level perspective of topology [15].

Latency and Jitter: In general all applications benefit from less end-to-end latency and jitter. However, its effect on users depends on the nature of application. For example, non-realtime applications such as browsing applications are more resilient to packet level latency and jitter compared to real-time interactive applications like VoIP or augmented reality. Considering the increasing number of devices such as google-glass or Vehicular Ad-hoc Network (VANET) based applications where response time is critical, it is desirable to deliver services from network-edge.

Smart Pipe for Operators: It is well known that the value of the Internet is gradually shifting from operators to Over-The-Top (OTT) Application Service Providers (ASP). Although operators have followed suit by replicating similar services, they have been unsuccessful due to many reasons which include their own core business focus, lack of global footprint, and affinity by users to established ASPs. Another response to this situation is to leverage their own infrastructure footprint to enable ASP's services to serve their subscribers at scale with significantly better Quality of Experience (QoE) compared to the situation today with well understood Service Level Agreements (SLA). This also requires openness from operators to allow service delivery control to ASPs, as they require direct control over their services, subscribers, and contents while agreeing on a revenue sharing model with the operators.

The objective of this paper is to provide an overview of the ICN based edge service platform proposed in [12] as discussed in Section II. Compared to [12], this paper introduces the stages of service composition which are discussed in Section III. Section IV provides details of our ICN based edge service prototype.

II. ICN Edge-Service Framework

An end-to-end view of our framework is shown in Figure 1. Future networks based on NFV shall be an interconnection of edge-clouds that leads to orchestration of intra-cloud as well as inter-cloud services to manage computing, storage, and bandwidth resources shared among heterogeneous services. These edge-cloud realizations are expected to be deployed in the vicinity of operator's CO or at PoP.

Considering incremental deployment, the network infrastructure of the ICN platform is realized as ICN router hosting ICN services (we call this ICN Service Router (ISR)) overlaid over current IP network, with a service layer for dynamic service management. All the ICN platform components can be realized as virtual functions; however for performance reasons the ICN router is non-virtual as it performs ICN functions of name-based routing and caching of the aggregate service traffic. The important functional components of the platform are: *ICN service platform*, *ICN service orchestrator*, and *ICN service API*.

ICN Service Platform: The ICN service platform is composed of the ISR and the virtualized services to handle platform and service specific functions. The ISR implements an ICN forwarding plane which conducts name-based routing inter-connecting distributed ICN services and leverage ICN features like content multicasting, caching, and content-based security. The platform specific service function includes *ICN Service Access Point* (ICN-SAP) per ISR and a global *ICN Service Profile Manager* (ICN-SPM). The counterpart of ICN-SAP is the *ICN Service Access Layer* (ICN-SAL) which resides in the User Entity (UE); it enables UEs to connect to the nearest ICN-SAP instance. The ICN-SAP interprets ICN-SAL requests and invokes service management functions such as discovery, resolution, publishing, or adaptation to context changes over the ICN UNI-API. ICN-SPM is a centralized database of service profiles that holds static and dynamic information such as service statistics (e.g. number of active service instances and its reachability, popularity), and SLA requirements. The SPM can be queried by any other services for its own purpose. The SLA objectives can be further protected by access policies in the service owner.

ICN Service Orchestrator: The service orchestrator interfaces with the ICN service owners (operator or third party) to program the ICN service platform's resources through the *ICN S-UNI*. This interaction invokes relevant *ICN service controller* and *ICN network controller* functions to meet service objectives. The S-UNI allows service owners to express service requirements during provisioning and request dynamic changes to computing/storage/bandwidth resources to adapt to service load conditions. The service orchestrator converts service requirements to compute storage, service connectivity, and bandwidth requirements to meet service objectives. This API also provides service monitoring capability to service owners benefiting from the ability to scale services.

The *ICN service controller* handles the functions to scale compute and storage (caching) in response to such requests by the service orchestrator. The *ICN network controller* plays the role of an SDN controller but works on network abstractions based on ICN semantics which can vary from conducting name-based routing, cache management, context handling, and service composition logic. The network controller interfaces with per-service controller instances to provide bi-directional flow of fine grained service flow events and responses to and from the ICN service platform.

ICN Service API: APIs can be categorized into those interfacing external entities such as the user and the ASPs for service access and management and those required internally to promote inter-operability. The service APIs include the ICN A-UNI and S-UNI. The ICN A-UNI includes standard APIs used by applications for service management and for SAL-SAP (discussed next) interaction. The ICN S-UNI implements service management functions to enable service control and monitoring for the service owners. The ICN platform-API represents the set of internal APIs to allow standardized interaction between distributed SAP instances, and ICN service orchestrator and the ISR for ICN service and network virtualization. For e.g., the inter-SAP interaction can realize contextual changes triggered over SAL-SAP interface such as seamless mobility as users move from one access point to another. The ASP customers of the ISP interacts over the S-UNI interface. We next discuss various service features enabled over the SAL-SAP interface.

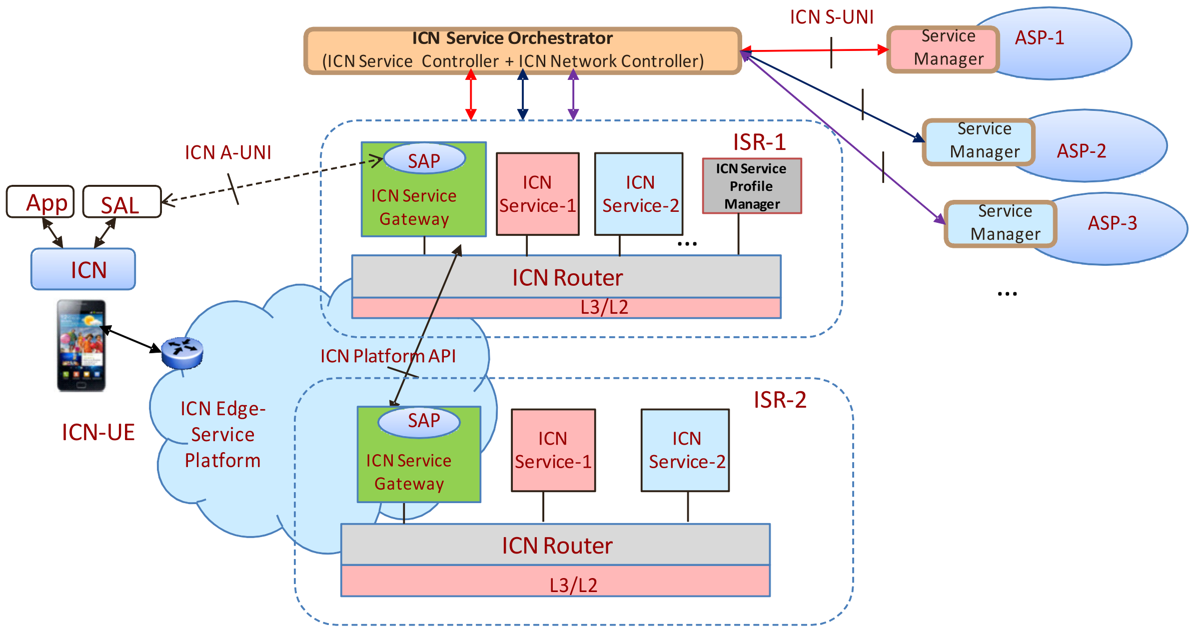


Fig. 1. ICN based edge service framework.

A. SAL-SAP Interaction

A high level view of the service layer to enable ICN-based edge services is shown in Figure 1. This figure shows the key control elements enabling edge service framework along with open-API between UEs, ISP, and ASPs and focuses on the important functions offered by SAL and SAP which aids contextual interaction between users/applications and the ICN service platform.

Following are the key functions supported as part of the SAL-SAP (A-UNI API) interface:

Service Discovery: Users and applications learn about services in their vicinity by conducting service discovery over the A-UNI API. When an application requests the SAL to discover services, this invokes the discovery API over A-UNI API. Once SAP receives the query, it would query the ICN-SPM for active services matching the request and policies and responds with the appropriate service-ID, and location semantics.

Service Request Management: Applications leverage SAL to generate service management requests ranging from service attachment to context change notifications. The contextual support include but is not limited to mobility, location, connectivity, or social context. Depending on the service, the context change itself could be managed independent of the application such as in the case of mobility where the underlying service connectivity can be re-programmed by SAL on behalf of the application, or the application explicitly can request such adaptation as a result of user action.

Service Publish: With cheaper storage and computing and movement towards IoT, it is very likely that services will be provided by many end devices, or using service proxies for the constrained environments such as sensor networks. SAL allows users/applications to publish their own services for remote consumers via A-UNI interface or publish services via S-UNI to make them available to other ISRs. Published services can be instantiated and executed at the edge ISR. Such publishing should preserve the requirements of the services related accessibility restrictions, and also scale considering the potential number of service end points. Publishing end host services requires control plane interactions to obtain configuration information for naming services and service registration to enable resolution to local service instances. Further policy enforcement for the service can be conducted at producing end point, service proxy, or even in the network with the trade-off of a more complex ICN forwarding plane.

The intelligence of our platform lies in services reacting to contextual changes of users. While these context changing events can be handled by the service controller, it has to be handled in a way that satisfies the service SLAs. In the following, we provide a discussion on context representation, expression and how it triggers service composition and executed over an ICN-based service framework.

III. ICN Service Composition at the Edge

Service composition in the context of this work is a more generalized vision that incorporates the ICN perspective which mimics the higher level view of service function interactions. The objective of service composition is to meet the SLA requirements of services and applications and be capable of coping with the dynamics of wireless networks in terms of connectivity, data rate fluctuations, mobility of users, or service function migration. Next generation service overlay networks can benefit from the elasticity of resources in the network enabled edge-based systems. Network enabled cloud based solutions bring cloud capabilities into the underlying network infrastructure to enable deployment of services and applications wherever and whenever needed. It is therefore important to design the communication systems in a way that can adapt to changes of applications and services requirements. Rapid development in the cloud computing domain and convergence of networking and compute resources by virtualization facilitates migration of many applications and services to the cloud. Customized services to meet the user requirements need to be composed by available components offered by different vendors. A distributed QoS management at the network level can be achieved by virtualized overlay network functions that can drive dynamic reconfiguration at the host and network level (using protocols like OpenFlow) to meet service requirements. The objective of service

978-1-4799-6390-4/15/$31.00 ©2015 IEEE

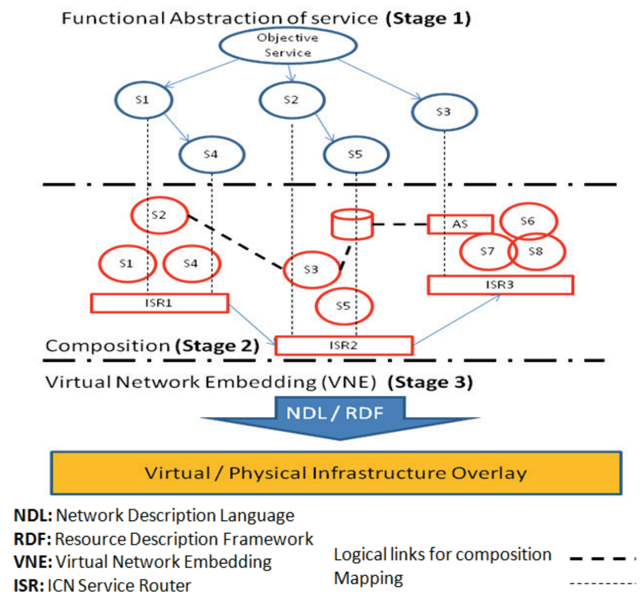


Fig. 2. Stages of service abstraction towards infrastructure embedding.

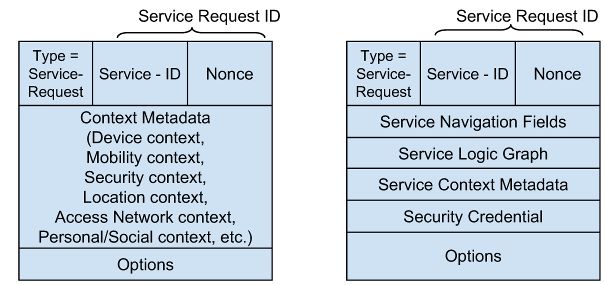
composition is to map high-level needs of services and applications and the SLA requirements to an end-to-end SLA requirement to be met by service components that form the composed service. The SLA requirements will then be translated into virtual network infrastructure QoS requirements that can be provisioned, monitored and reconfigured upon violation of SLA [16].

The process of service composition as described in more details in [16] involves service abstraction and identification, service composition and specification, and service implementation and realization.

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 service 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.

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. In addition, actions can be triggered from the running service side such as enforcing different location sensing, initiating a vertical handover, enabling or disabling a camera and etc. Contextual information can be managed at the core SAP and in part at the UE end by a *context management module* [14], [16].

Protocol realization: We propose a generic extended service request and response model to enable this service level interaction. This protocol can be realized as an ICN overlay i.e. tunneled between ISRs or as an underlay, i.e., network layer aware as proposed by authors in [13]. Figure 3 (a) and (b) present the request PDU formats on the UE side and the network side. As shown in this figure, `Type` field indicates whether the packet is a service request or response; `Service-ID` identifies service's persistent service identifier; and the `nonce` is a random number that is appended to the Service-ID to logically form a unique *Request-ID*. This is followed by consumer's context metadata information as described earlier. In addition, service request and response could also include consumer/producer related security credential information, for features like service authorization and access control.



(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

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. 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.

A. Service Abstraction and Identification

The abstraction of the service is the result of extracting information such as type of user device, preferences and location, from the service request PDU. Therefore, prior to the service abstraction and identification, processing of the the service request is required. In this stage, required components to form a service are identified based on the available published description of the service and type of service that be offered as well as the SLA offered by that service. 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.

B. Service Composition

This stage involves the generation of the service logic graph from abstract definition of the previous stage. Managing the service logic processing and creating the service logic graph from the abstract level of service definition is a critical part of service orchestration. 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. 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

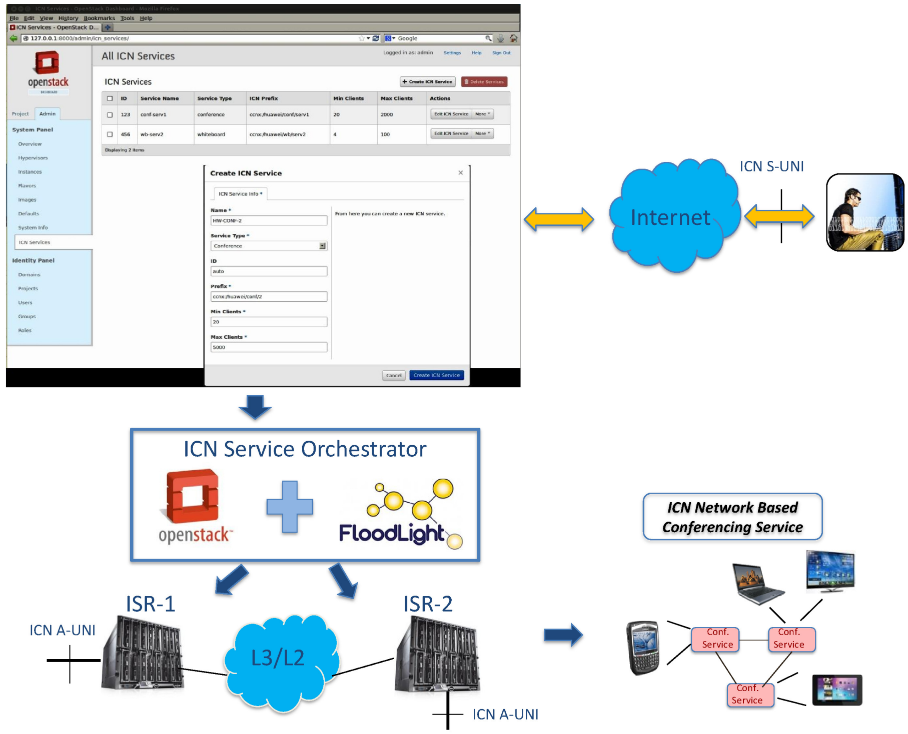


Fig. 4. Prototype of network based conferencing service over ICN.

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 described earlier 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) 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 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.

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) [7]. 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 consists allocation of virtual entities meeting the SLA requirements and minimizing the usage of resources. 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 are 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. The existing algorithms can be categorized into static vs dynamic, centralized vs distributed and concise vs redundant. For the purpose of our work, a dynamic, distributed and concise approach is of interest such as [8] where authors propose a distributed self organizing model to manage the substrate network resources. For the VNE problem we consider an embedding optimization problem with a cost objective that is a function of delay and compute resources

978-1-4799-6390-4/15/$31.00 ©2015 IEEE

and solve the problem with a Greedy algorithm approach. The VNE stage is part of our work in progress [16] and is not yet implemented in the prototype.

IV. ICN Based Edge Cloud Prototype

In this section, we validate the concepts described in this paper by demonstrating the prototype that is developed in our research laboratory. The prototype setup is shown in 4, and has been demonstrated as Virtual Service Edge Router (VSER) platform [4]. The ISRs are realized over a commodity platform, the Dell's PowerEdge M1000e blade servers. This chassis has two type of servers, a high resource rich server with 12 computing cores at 2.9Ghz and 128GB of RAM hosting the ICN router, and servers with same cores at 2.5Ghz and 32GB RAM to host virtual service instances. The ICN router is based on CCNx [1] with focus on simplifying current encoding format and parallelized implementation exploiting server's computing pool and memory resources. The service orchestrator extends OpenStack [3] and floodlight [2] to provision ICN services and conduct service routing.

In [12] we proposed a network based conferencing framework over CCN comprising of client-agent on UE, and proxy-agent and a controller to handle many participants in a scalable manner. The proxy-agents are distributed local instances, while the controller is centralized coordinating the proxy-agents. The objective of this conferencing framework is high speed synchronization of new content among the participants. Variant of this technique can be used to handle audio/video/text media components of a conferencing session.

Figure 4 shows a web interface that inputs information such as number of participants per-site, conference ID, UE device information. In the current setup, workload is determined by making simple assumption of estimated traffic per instance considering text messaging traffic. However, this can be more complex considering device types and media (voice, text, video) involved, and contextual information such as mobility, or service popularity from real time monitoring. When the conference manager provides this information, the service orchestrator provisions the required number of VMs. Upon this, the ISRs notify the ICN network controller about these active service instances (the ISR level topology is also discovered and managed by the ICN network topology manager). The network controller then notifies the corresponding ICN service controller, in this case the conference service controller. The controller conducts name-based service logic routing to interconnect the conference service framework components (client-agent, proxy-agent, and controller) instantiated to handle the conference. The participants join the conference by first learning about the service through the SAL, which also programs the CCN FIB policies in the UE to enable the conferencing application to connect to the closest ISR's proxy-agent. When a participant joins, it learns about the current participants, and conference framework helps the participant synchronize the conference state. The participants then express interests to the ISRs to obtain the latest data generated by the remote participants.

This prototype currently handles new participants joining and leaving, and is being evolved to handle contextual factors like device type and to also include voice and video media type. Furthermore, the platform itself is generic enough to handle any other type of service.

V. Conclusion

ICN paradigm allows applications, services and networks to interact using naming primitives that decouples applications from the transport layer. ICN has been mostly seen in the context of content distribution; this paper elevates ICN as a service platform capable of hosting dynamic heterogenous services, and enabling rich context-driven service interaction between consumers and services . This builds on ICN semantics which augments naming semantics with contextual parameters expressible by consumers, and orchestrated by leveraging affordances offered by NFV and SDN. In this article we proposed a novel design to realize ICN based network-edge services through a configurable and context-aware platform allowing ISP and ASP interaction for end-to-end service delivery. Such a platform elevates operator visibility from providing connectivity to being integral part of the service chain. We validate this concept through a lab prototype and demonstrates it realization through an ICN-based conferencing service over the proposed service platform.

Acknowledgment

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

References

[1] Project CCNxTM. http://www.ccnx.org.
[2] Project FloodLight. http://www.projectfloodlight.org/floodlight/.
[3] Project OpenStack. http://www.openstack.org.
[4] A. Chakraborti, V. Rajaraman, S. Zhao, A. Azgin, R. Ravindran, and G. Wang. Multi-party conference over virtual service edge router (vser) platform. In *ICN SIGGCOMM*, pages 205–206, 2014.
[5] M. Chiosi, D. Clarke, J. Feger, C. Cui, J. Benitez, U. Michel, K. Ogaki, M. Fukui, D. Dilisle, I. Guardini, D. Lopez, F. Ruhl, P. Sen, and et al. Network functions virtualisation: An introduction, benefits, enablers, challenges & call for action. In *SDN and OpenFlow World Congress*, 2012.
[6] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis. The Locator/ID Separation Protocol (LISP). *draft-ietf-lisp (Experimental RFC)*, 2013.
[7] A. Fischer, J. Botero, M. Beck, H. De Meer, and X. Hesselbach. Virtual network embedding: A survey. 2013.
[8] C. C. Marquezan, L. Z. Granville, G. Nunzi, and M. Brunner. Distributed autonomic resource management for network virtualization. In *Network Operations and Management Symposium (NOMS), 2010 IEEE*, pages 463–470. IEEE, 2010.
[9] 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.
[10] S. Paul and R. Jain. Openadn: Mobile apps on global clouds using openflow and software defined networking. In *Globecomm*, December, 2012.
[11] 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.
[12] 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.
[13] M. Sifalakis, B. Kohler, C. Scherb, and C. Tschudin. An information centric network for computing the distribution of computations. In *ICN, SIGGCOMM*. ACM, 2014.
[14] 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.
[15] P. TalebiFard and V. Leung. Expansion properties of topology for networking of information in cloud. *Transactions on Parallel and Distributed Systems (TPDS)*, 2013.
[16] P. TalebiFard, R. Ravindran, A. Chakraborti, G. Wang, and V. Leung. Towards a context adaptive icn-based service centric framework. In *Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), 2014 10th International Conference on*, pages 180–185. IEEE, 2014.
[17] 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 PP, pages 1–26, 2013.

978-1-4799-6390-4/15/$31.00 ©2015 IEEE