Multi-party Conference over Virtual Service Edge Router (VSER) Platform

Asit Chakraborti†, Vinodkumar Rajaraman‡, Shuai Zhao§, Aytac Azgin†, Ravishankar Ravindran†, and Guoqiang Wang†
†Huawei Research Center, Santa Clara, CA, USA.
{asit.chakraborti, aytac.azgin, ravi.ravindran, gq.wang}@huawei.com
‡Infinite Computer Solutions, Chennai, India. vinodkumarr@infinite.com
§University of Missouri-Kansas City, MO, USA. shuai.zhao@mail.umkc.edu

ABSTRACT

Realizing large scale multi-party conference is a challenge today when realtime and high bandwidth multimedia components are involved due to lack of scalability of server and bandwidth resources. We demonstrate a scalable conference design over the Virtual Service Edge Router (VSER) platform which is an ICN edge service router with the capability of hosting arbitrary realtime and non-realtime services as virtual machines (VM). The platform services are orchestrated through a programmable framework and takes advantage of scalable forwarding plane for content distribution.

1. INTRODUCTION

Scalability challenges of IP based multi-party conference solutions when bandwidth consuming media streams such as video are involved has been shown in many studies, e.g. [2]. In server based solutions, the server is the bottleneck; while P2P based solution suffer from the per-participant processing and uplink bottleneck constraint. Studying different commercial systems, [2] observes server based systems offer better performance compared to P2P architectures and scale to a maximum of 20 active participants, albeit with poor overall QoE. ICN (CCN [1] in our case) can address bandwidth scaling through network-level content abstraction over per-participant namespace, thereby enabling large scale multicast of user's media. The service scaling is addressed by the Virtual Service Edge Router (VSER) platform comprising of distributed CCN-based service edge routers managed by a service orchestrator based on NFV and SDN frameworks. A problem from applying CCN directly to realtime conference is the lack of knowledge of a producer's latest content name by the consuming participants. Chronos [4] addresses this through a serverless digest synchronizing mechanism that suffers from multiple simultaneous updates and failure recovery issues; also this proposal doesn't naturally support realtime audio or video sessions.

ICN'14, September 24–26, 2014, Paris, France.
ACM 978-1-4503-3206-4/14/09.
http://dx.doi.org/10.1145/2660129.2660137.

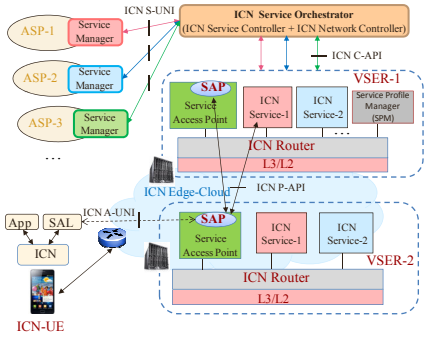

Figure 1: VSER platform architecture.

We propose a *Push* based solution on the VSER platform. To aid multi-party conference, service instances are realized that help synchronize content *fingerprints* (*i.e.* unique name component suffixes of the *content*-ID) among participants, where a participant maps to one of these service instances over VSER. Once the fingerprint is learnt, the content is *Pulled* by the consumer through the CCN forwarding plane. While this approach scales to many participants for text-based chat with generous QoE requirements, the synchronization latency is too high for participants generating realtime audio or video stream. We adapt our scheme on such streams by enabling periodic notifications over multiple encoded audio-video frames, and taking advantage of Interest pipelining to retrieve them. Recovery from user entity's (UE) transient or long term failures, discussed in [3], is also easy to handle in this model as only two entities, *i.e.* the UE and the service instance are involved during the recovery.

Service provisioning and name-based routing between these VMs are controlled by conference-specific service and network controllers. In general, the VSER platform can host any service ranging from content distribution, conferencing, or an IoT application controlled by respective service controller application.

2. CONFERENCE OVER VSER PLATFORM

The VSER platform is envisioned as a service hosting CCN edge router to enable service contextualization (*e.g.* mobility, user preference adaptation) and customization (*e.g.* geography preference, service scaling) through Open-APIs to consumers and application service providers (ASP). The

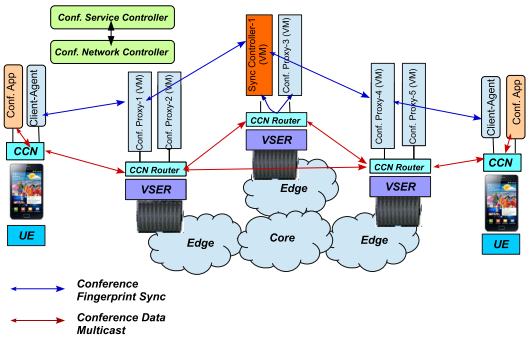

Figure 2: VSER based conference architecture.

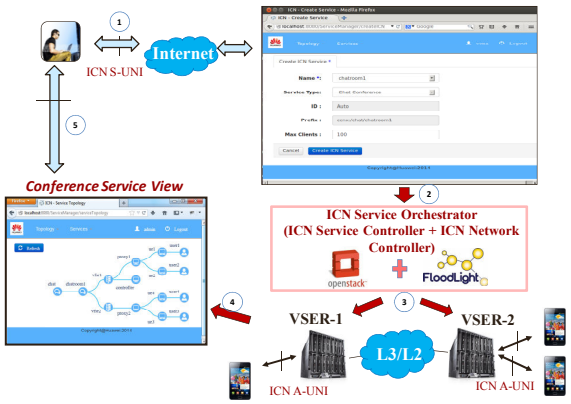

Figure 3: VSER platform based conference demo.

VSER platform shown in Fig. 1 is discussed in [3]. The key components of the architecture are the service access layer (SAL) on the UE that aids service discovery and service context expression; service access point (SAP) per VSER node to help UE applications connect to services, and help device or service context adaptation; global service profile manager (SPM) manages the database of active services; VSER forwarding plane is based on CCN; and the ICN service orchestration layer with service and network controller components to conduct service provisioning and dynamic name-based routing based on factors affecting the service.

The conference solution proposed in [3] is shown in Fig. 2. Here participants are connected to distributed VSER nodes and serviced by two service components (realized as VMs): *Conference Proxy* (*cPrx*) and the *Sync Controller* (*sCon*). To aid chat application sync its fingerprints, *client-agent* in the UE helps push them to the serving *cPrx*. *sCon* and *cPrxs* forms a hub-and-spoke topology to support faster synchronization and recovery from transient failures. The provisioning of *cPrx* and *sCon* is managed by a *conference service controller*, and routing between client-agent and *cPrx*, among *sCon* and *cPrx* instances, and the UEs is managed by the *conference network controller*. We evaluated the scalability performance of the architecture in [3] using simulations. The results on convergence time for the multi-party *video* conference for varying number of participants is shown in Fig. 4(a). Here the participants are equally distributed among the *cPrx* instances. The two plots correspond to the convergence among consumers local to the producer hosted by the same *cPrx* and those hosted by remote *cPrxs*.

3. CONFERENCE DEMO

The demo is realized on two Dell PowerEdge M1000e chassis, each with a high end blade (12 cores, 128GB) for CCN forwarding and two lower end blades to host the VMs. The forwarding engine follows the CCNx1.0 protocol specification [1]. The demo set-up shown in Fig. 3, begins by the user (ASP) providing conference provisioning parameters such as conference service namespace, chat room name, and the maximum number of users to the conference controller interface that is normalized to the number of conference service functions (*cPrx* and *sCon* (VMs)) to be provisioned in the VSERs. The provisioning event is notified to the conference network controller that in turn programs VSER's FIB to inter-connect *cPrx* and *sCon* functions. The UE then discovers the provisioned chat rooms through the SAL by querying the SAP. This information is notified to the application which joins a chat room through a user initiated action. The join action results in learning the control namespace, the list of active participants and their corresponding participant-specific content namespace to allow participant interaction. The smart clients interact through rich text, audio, and video.

Extended OpenStack and Floodlight realize the ICN conference service and network controller. A conference application controller over the network controller has a real-time view of conference topology such as the mapping of *cPrx* and *sCon* VMs to the VSERs, and participant list corresponding to each *cPrx* instance. Similar parallel views exist when multiple simultaneous conference sessions are provisioned. The prototype is evaluated by emulating up to 48 participants. Fig. 4(b) shows relative invariance of convergence time for varying number of active chat participants distributed over two VSER nodes hosting a *sCon* and two *cPrx* instances.

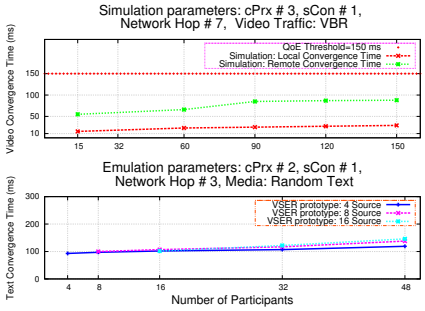

Figure 4: VSER convergence performance: (a)simulation, (b)emulation

4. REFERENCES

[1] CCNx1.0 protocol specification, `http://www.ietf.org/mail-archive/web/icnrg/current/pdfZyEQRE5tFS.pdf`.

[2] Y. Lu, Y. Zhao, F. Kuipers, and P. Van Mieghem. Measurement study of multi-party video conferencing. In *IFIP Networking*, 2010.

[3] R. Ravindran, X. Liu, A. Chakraborti, X. Zhang, and G. Wang. Towards software defined icn based edge-cloud services. In *CloudNet*, 2013.

[4] Z. Zhu, C. Bian, A. Afanasyev, V. Jacobson, and L. Zhang. Chronos: Serverless multi-user chat over ndn. Technical report, NDN-0008, 2012.