Demo: ACACIA – Context-aware Edge Computing for Continuous Interactive Applications over Mobile Networks

Junguk Cho[†], Karthikeyan Sundaresan[‡], Rajesh Mahindra[‡], Jacobus Van der Merwe[†], and Sampath Rangarajan[‡]

University of Utah[†]

NEC Laboratories America, Inc.[‡]

junguk.cho@utah.edu, karthiks@nec-labs.com, rajesh@nec-labs.com kobus@cs.utah.edu, sampath@nec-labs.com

ABSTRACT

We propose ACACIA- a service abstraction framework that enables continuous interactive (CI) applications on edge clouds in mobile networks. We will demonstrate the ACACIA architecture and illustrate its feasibility by using an augmented reality application as an example use case.

CCS Concepts

•Networks → Mobile networks; Cloud computing;

Keywords

mobile network; service offerings; context-aware edge computing; NFV/SDN

1. INTRODUCTION

Future continuous interactive (CI) applications, like augmented reality (AR), virtual reality (VR), etc., will require very low end-to-end latencies and computational resources close to the network edge [2]. However, current 4G mobile networks are incapable of providing such delays since the core mobile network itself incurs a substantial delay (high tens of milliseconds or more) [3].

There is widespread expectation that the emerging 5G mobile network will address these concerns. One potential solution is using cloud platforms in the form of mobile edge computing (MEC) to meet edge computing requirements, like low latencies and computation resources. In addition, localization and context discovery frameworks significantly help CI applications to be realized. Indeed, to realize CI applications, many necessary components have been proposed and studied in standalone manner. However, there has been no an end-to-end combination of these components to satisfy CI application requirements over mobile networks. Especially, the question of how these components would be combined into service offerings by mobile network carriers, have not been meaningfully addressed. To address this problem, we propose the design of a service abstraction framework called ACACIA that enables CI applications on edge clouds in mobile networks. ACACIA leverages client context information through proximity service discovery to optimize network and application processing and combines SDN/NFV principles with LTE/EPC QoS mechanisms to offload CI application traffic to mobile edge clouds on demand.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

MobiCom'16 October 03-07, 2016, New York City, NY, USA

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4226-1/16/10.

DOI: <http://dx.doi.org/10.1145/2973750.2985623>

In designing ACACIA our key insight is that a user's context with respect to her environment is a critical factor in optimizing both the network and application aspects required by MEC-based CI applications. Specifically, a CI application using MEC-based services should be directed to an edge cloud instance in close proximity to the user where the required services are available (i.e., providing network level optimization). Similarly, the specific context of the user can be used to inform and optimize the application functionality. For example, A retail shopping AR application can benefit, not only from knowing the relative location of a user (e.g., the appliance department), but also from knowing that the user is interested in a specific type of appliance (e.g., a microwave oven).

ACACIA leverages this insight to realize *Context-Aware MEC for CI applications over mobile networks*. Specifically, ACACIA leverages user context in the form of their interest in specific CI proximity “services” as well as proximity to landmarks to (i) create *on-demand* connections to MEC servers only when a client is interested in CI applications and its service is available, while maintaining always-on connection to the core network, thus alleviating the control overhead in establishing such additional connections, (ii) automatically *classify* MEC traffic at the source (client) itself, and *re-direct* only the MEC traffic to the closest MEC server, thereby avoiding the need for traffic inspection and classification, and (iii) speed-up the application processing through *location-aware optimization*, that is especially useful in CI services like AR/VR. To our knowledge, ACACIA represents the first holistic end-to-end design to support CI applications over mobile networks.

2. ACACIA DESIGN

Figure 1: ACACIA Overview

Figure 1 shows an overview of the ACACIA mobile edge computing (MEC) service framework for context-aware continuous interactive (CI) applications. As shown, ACACIA assumes a core

mobile network consisting of both centralized network functions (e.g., using LTE nomenclature, standard evolved packet core (EPC) functions), as well as distributed mobile edge cloud (MEC) instances. Following current technology trends, the MEC instances, in addition to hosting application servers, are capable of realizing virtualized core network functions (i.e., NFV/SDN-based EPC functions). Finally, ACACIA uses the D2D-based LTE-direct proximity discovery functionality.

Given this context, ACACIA enables context-aware CI applications by orchestrating the following functional components: (i) *User context discovery*: The ACACIA device manager enables applications on mobile devices to make use of the publish-subscribe capabilities of LTE-direct proximity services to advertise (publish) and learn about (subscribe to) the availability of “services” that are of interest to users. The semantics of the messages associated with these service advertisements are application specific, but an “interest match” by a subscriber will trigger a CI application to provide the user with more information. (ii) *Context-aware traffic redirection*: When CI applications require mobile edge cloud resources to realize their functionality the ACACIA device manager will establish connectivity to the mobile edge cloud. The ACACIA device manager interacts with EPC functions in the mobile core network, which makes use of the proximity service discovery information to establish connectivity to the appropriate CI server on a mobile edge cloud that would satisfy the latency requirements of the CI application. (iii) *ACACIA Mobile Edge Network*: ACACIA combines existing LTE/EPC QoS bearer framework with novel SDN and NFV enabled mechanisms to realize the standards-compliant MEC data-plane offloading for only CI traffics. (iv) *Context-aware application optimization*: With connectivity to the mobile edge cloud in place, the ACACIA device manager provides location information to the CI application, which can be used to optimize its functionality. Thus, ACACIA is capable of enabling a variety of CI related use cases.

3. DEMONSTRATION OVERVIEW

Figure 2: ACACIA prototype and demo setup

Figure 2 shows the ACACIA prototype and demonstration setup. We use an Android smartphone (Nexus5) as User Equipment (UE) and a commercial LTE small cell to realize the radio access network (RAN). We use two laptops to run an NFV realization of both the MEC and the mobile core network (i.e., EPC). One laptop runs GW-Cs, GW-Us and other EPC components, as well as an SDN controller, representing an SDN-enabled EPC network. This facilitates a default bearer from the UE to the Internet. The other runs an MEC instance, including GW-U functionality, to enable offloading between the UE and CI server in the MEC (via an on-demand dedicated bearer). We use OpenEPC for the EPC components, Open vSwitch for GW-U and a Ryu SDN controller to control the network.

In this demonstration, we showcase an AR application (i.e., an object detection application) as an ACACIA use case. The AR application running on the UE (smartphone) continually sends frames from its camera to a MEC-based CI server. The CI server analyzes each frame and matches it against an object database. If there is a match, it returns information related to the object like price, reviews, customer ranking etc. The AR application augments the camera view with this information.

In this demonstration we will show the following (Note that due to limitations of the LTE eNodeB we use in the demonstration we do not show LTE-direct functionality of ACACIA (i.e., user context discovery and context-aware application optimization)):

- On-demand user context based (i.e., user interest or application triggered) network provisioning for CI application.
- Selective traffic offloading to a CI server by using the LTE QoS framework and SDN mechanisms (i.e., split-mode GWs).
- Lower latency of the MEC than existing LTE/EPC networks

With reference to Figure 2, the demo scenario is as follows:

① Initially, the UE is connected to the mobile network via a default bearer.

② *Request*: When the user uses the CI application, the ACACIA device manager sends a request to the MEC Registration Server (MRS) (using the default bearer), to setup a dedicated bearer. The request includes the UE’s IP address and service information.

③ *Create*: If the request is successful, the MRS triggers the PCRF (Policy Control and Charging Rules Function) with the service ID, and flow information including the IP addresses of UE and CI server corresponding to service information. PCRF dynamically generates policy rules and invokes PCEF (Policy and Charging Enforcement Function) in PGW-C with the policy rules, which results in a network-initiated dedicated bearer activation [1].

④ *Set-up*: The PGW-C, SGW-C, MME, eNB, and UE then exchange control messages to setup a dedicated bearer. To set up the data plane for the dedicated bearer to the MEC, the PGW-C and SGW-C provides IP addresses of *local* GW-U (not those of GW-U serving default bearers) as the F-TEID (Fully Qualified Tunnel Endpoint Identifier) values in the *Create Bearer Request/Response* control messages. With this, the MME, eNB and UE all set up the requested dedicated bearer.

⑤ *Route*: After the successful exchange of control messages, the GW-Cs send information to the Ryu SDN controller. It inserts SDN (OpenFlow in our setup) flow rules into *local* SGW-U and PGW-U based on information such as TEID, and IP address of GW-U and eNB from control messages.

⑥ *Running AR App*: At this point the AR application communicates with MEC-based CI server via the dedicated bearer. When there is a match in the CI server, the smartphone augments the camera view with information received from the CI server.

4. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under grant numbers 1343713 and 1305384.

5. REFERENCES

- [1] 3GPP TS 23.401.
<http://www.3gpp.org/DynaReport/23401.htm>.
- [2] M. Satyanarayanan. A brief history of cloud offload: A personal journey from odyssey through cyber foraging to cloudlets. *GetMobile: Mobile Comp. and Comm.* 2015.
- [3] K. Zarifis et al. Diagnosing path inflation of mobile client traffic. In *Proceedings of the 15th International Conference on Passive and Active Measurement*, PAM 2014.