RTGWG Working Group
INTERNET-DRAFT
Intended Status: Informational
Expires: May 7, 2020

L. Geng
China Mobile
P. Willis
BT

November 4, 2019

Compute First Networking (CFN) Scenarios and Requirements
draft-geng-rtgwg-cfn-req-00

Abstract

Service providers are exploring the edge computing to achieve better response times and transfer rate by moving the computing towards the edge of the network in scenarios like 5G MEC, virtualized central office, etc. Providing services by sharing computing resources from multiple edges is emerging. The service nodes from multiple edges normally have two key features, service equivalency and service dynamics. When the computing resources attached to a single edge site is overloaded, static service dispatch can possibly keep allocating the service request to it and cause inefficient utilization. The service request to edge computing needs to be dispatched to and served by the most suitable edge to improve user experience and system utilization by taking both the available computing resources and network conditions into account.

This draft describes scenarios and requirements of Compute First Networking (CFN) to make the computing and network resource to be considered in a collaborative way to achieve a more balanced network-based distributed service dispatching.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

Copyright and License Notice

Table of Contents

1. Introduction

Edge computing aims to provide better response times and transfer rate by moving the computing towards the edge of the network. Edge computing can be built on industrial PCs, embedded systems, gateways and others. They are put close to the end user. There is an emerging requirement that multiple edge sites (called edges too in this document) are deployed at different locations to provide the service. There are millions of home gateways, thousands of base stations and hundreds of central offices in a city that can serve as candidate edges for hosting service nodes. Depending on the location of the edge and its capacity, each edge site may have different computing resources to be used for a particular service. The computing resources hosted on an edge is limited. At peak hour, computing resources attached to the closest edge site may not be sufficient to handle all the incoming requests. Longer response time or even request dropping can be experienced by the user. Increasing the computing resources hosted on each edge site to the potential maximum capacity is neither feasible nor economical.

At the same time, with the new technologies such as serverless computing and container based virtual functions, service node on an edge can be easily created and terminated in a sub-second scale. It makes the available computing resources for a service change dramatically over time at an edge.

The traditional method of static pre-configuration of which service request is dispatched to which edge causes the workload distribution to be unbalanced in terms of network load and computational load. The reason is there is no interaction on scheduling capability between edges about the hosted computing nodes. When computing resources on one edge are overloaded or even unavailable, the requests may still keep coming and cause the service experience deteriorates. Most current solutions use the application layer functions to solve this issue. It requires L4-L7 handling of the packet processing, such as reverse proxy, which takes longer connection time. It is not an efficient approach for huge number of short connections.

Multi-site edge computing has the distributed nature. Service providers are starting to build the edge platform to allow a large number of requests to be served by sharing the computing resources from service nodes at multiple edges in a collaborative way. That is to say, a service request potentially can be handled by different service nodes located in different edges and it has to be decided which one is the most appropriate to serve this request in real time. Such an approach can improve the system utilization to serve more end users by balancing the workload distributed to multiple edges intelligently. Intelligence here means considering both the network

conditions and available computing resources.

Both computing load and network status are treated as network visible resources, edge site can interact with each other to provide network-based service dispatching to achieve better load balancing. This is called Compute First Networking (CFN) in this document. It requires both network, edge and computing nodes to work coordinately for service selection dispatching between user to edge and edge to edge. Among them, edge to edge or inter-edge interaction is the focus of CFN in IETF related work. This draft describes usage scenarios and requirements of CFN.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Usage Scenarios of CFN

This section presents several typical scenarios which require multiple edge sites to interconnect and to co-ordinate with networks to meet the service requirements and ensure user experience.

2.1 Cloud Based Recognition in Augmented Reality (AR)

In AR environment, the end device captures the images via cameras and sends out the computing intensive service request. Normally service nodes at the edge are responsible for tasks with medium computational complexity or low latency requirement like object detection, feature extraction and template matching, and service nodes at cloud are responsible for the most intensive computational tasks like object recognition or latency non-sensitive tasks like AI based model training. The end device hence only handles the tasks like target tracking and image display, thereby reducing the computing load of the client.

The computing resource for a specific service at the edge can be instantiated on-demand. Once the task is completed, this resource can be released. The lifetime of such "function as a service" can be on a millisecond scale. Therefore computing resources on the edges have distributed and dynamic natures. A service request has to be sent to and served by an edge with sufficient computing resource and a good

network path.

2.2 Connected Car

In auxiliary driving scenarios, to help overcome the non-line-of-sight problem due to blind spot or obstacles, the edge node can collect the comprehensive road and traffic information around the vehicle location and perform data processing, and then the vehicles in high security risk can be signaled. It improves the driving safety in complicated road conditions, like at the intersections. The video image information captured by the surveillance camera is transmitted to the nearest edge node for processing. Warnings can be sent to the cars driving too fast or under other invisible dangers.

When the local edge node is overloaded, the service request sent to it will be queued and the response from the auxiliary driving will be delayed, and it may lead to traffic accidents. Hence, in such cases, delay-insensitive services such as in-vehicle entertainment should be dispatched to other light loaded nodes instead of local edge nodes, so that the delay-sensitive service is preferentially processed locally to ensure the service availability and user experience.

2.3 Cloud Virtual Reality (VR)

Cloud VR introduces the concept and technology of cloud computing and cloud rendering into VR applications. The end device usually only uploads the posture or control information to the cloud and then VR contents are rendered in the cloud. The video and audio outputs generated from the cloud are encoded, compressed, and transmitted back to the end device via high bandwidth network.

Cloud VR services have high requirements on both network and computing. For example, for an entry-level Cloud VR (panoramic 8K 2D video) with 110-degree Field of View (FOV) transmission, the typical network requirements are bandwidth 40Mbps, RTT 20ms, packet loss rate is 2.4E-5; the typical computing requirements are 8K H.265 real-time decoding, 2K H.264 real-time encoding.

Cloud VR service brings challenging requirements on both network and computing so that the edge node to serve the request has to be carefully selected to avoid the overloading.

3. Requirements

CFN in this document mainly targets at the typical edge computing

scenarios with two key features, service equivalency and service dynamics.

- Service equivalency: Equivalent service is provided by multiple edges to ensure better scalability and availability.

- Service dynamics: A single edge has very dynamic resources over time to serve a request. Its dynamics are affected by computing resource of service node, network path congestion, failover and others.

A service request should be routed to the most suitable edge for processing. The local edge is normally the first choice. At the same time, it is important to have the capability to route the request to the other edges when the local edge has insufficient computing resource or non-promising network path, depending on the service type and/or priority.

The following requirements need to be met for CFN,

- The service provided, or function called, should be identified in a format amenable to processing in the network

- Service request is sent in real time to the most appropriate one among all the service equivalent edges for processing. Such a request assignment should not be static. It should be based on the metrics for the service dynamics, including both network status and available computing resources.

- For applications that require flow affinity it must be possible to have a method to signal flow affinity requirements and handle flows on the same edge.

- Edge nodes may have limited compute resources therefore control and storage overhead must be minimised

4. Summary

CFN in this document tries to leverage the network distributed nature to help serve the edge computing requests in a more balanced way by considering both network status and computing resources among multiple edges. Inter-edge interaction is required in the dynamic service dispatching and network and computing resource information distribution.

This document illustrate some usage scenarios and requirements for CFN. CFN architecture should addresses how to distribute the computing resource information at the network layer, how the data

plane adapts when the edge to handle the first service request is not known in advance, and how to assure flow affinity.

5. Security Considerations

TBD

6. IANA Considerations

No IANA action is required.

7. Acknowledgements

The author would like to thank all participants who participated in the discussion of CFN at the earlier IETF/IRTF meetings.

8. References

8.1 Normative References

Authors' Addresses

Liang Geng
China Mobile
Email: gengliang@chinamobile.com

Peter Willis
BT
Email: peter.j.willis@bt.com