

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

Y. Li
J. He
Huawei Technologies
L. Geng
P. Liu
China Mobile
Y. Cui
Tsinghua University
November 4, 2019

Framework of Compute First Networking (CFN)
draft-li-rtgwg-cfn-framework-00

Abstract

Compute First Networking (CFN) leverages both computing and networking status to help determine the optimal edge among multiple edge sites with different geographic locations to serve a specific edge computing request. Requests for the same service can be determined and dispatched to different edges based on service requirements, network and computing resource conditions and other factors to achieve better load balancing and system efficiency. The request needs to be dispatched to the selected edge in real time and the subsequent packets from the same flow should be served by the same edge for flow affinity. This document describes a framework of CFN to achieve the desired features.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at

INTERNET DRAFT

Framework of CFN

Nov 2019

<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	CFN Framework	4
2.1	CFN Service Overview	5
2.2	Generic Workflow	7
3.	Control Plane and Data Plane	7
3.1	Control plane	7
3.2	Data plane	9
4.	Summary	12
5.	Security Considerations	12
6.	IANA Considerations	12
7.	Acknowledgements	12
8.	References	13
8.1	Normative References	13
8.2	Informative References	13
	Authors' Addresses	13

1. Introduction

Compute First Networking (CFN) scenarios and requirements document [[CFN-req](#)] shows the usage scenarios that require an edge to be dynamically selected from multiple edge sites with different geographic locations to serve an edge computing request based on computing resource consumption and network status in real time. For instance, edge site in residential area receives low request volume during working hours and high request volume during non-working hours. And the request volume received by the edge site in industrial park is the opposite. Such a pattern causes a big difference of computing load on different edge sites. Traditional static or hashing based service dispatch can not adapt to the unbalanced nature of computing load or rapid change of it on different edge sites. One edge such as the closest one to the client may have been overloaded and at the same time the other edges may still have plenty of computing resources to serve the requests. To efficiently leveraging the computing resources hosted on all edges, service requests should be dispatched and handled dynamically to make the computing and network resources consumed in a balanced way.

CFN assumes there are multiple service equivalent edges to serve a single service. A single edge has limited computing resources and different edges may have different resources available for serving a specific service at a specific time. In concept, multiple edges are interconnected and collaborated with each other to balance the service load in CFN. Computing resource available to serve a request is the top metric to be considered when dispatching a request. At the same time, the quality of the network path to an edge varies over time. CFN is a network based approach so that the request is dispatched to the optimal edge in terms of both computing resources available and network status on the fly.

This document presents a CFN framework which can support service equivalency and dynamics in edge computing to achieve better load balancing with no application dependency.

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.

CFN: Computing First Networking

[2](#). CFN Framework

Edge computing is expanding from a single edge site to networked and collaborated multiple edge sites to solve the issues of low efficiency and and low resource reuse. CFN enables large scale edge interconnection and collaboration, providing optimal service access and load balancing to adapt to service dynamics. Based on the real-time computing capacity available and the network conditions, CFN dynamically schedules computing request to appropriate service node, thus the resource utilization and user experience is improved.

Figure 1 shows the network topology of CFN. CFN node is the basic function entity in CFN network to provide the capability to exchange the information about the computing resource consumption information of service nodes attached to it and/or provide the CFN service access to the clients. Edge site (edge for short) is normally the site where the edge computing is hosted. CFN node can be a network virtual function (NFV) co-located with service node in a server. CFN node's function can also be provided by physical equipment like access router in access ring or metro network.

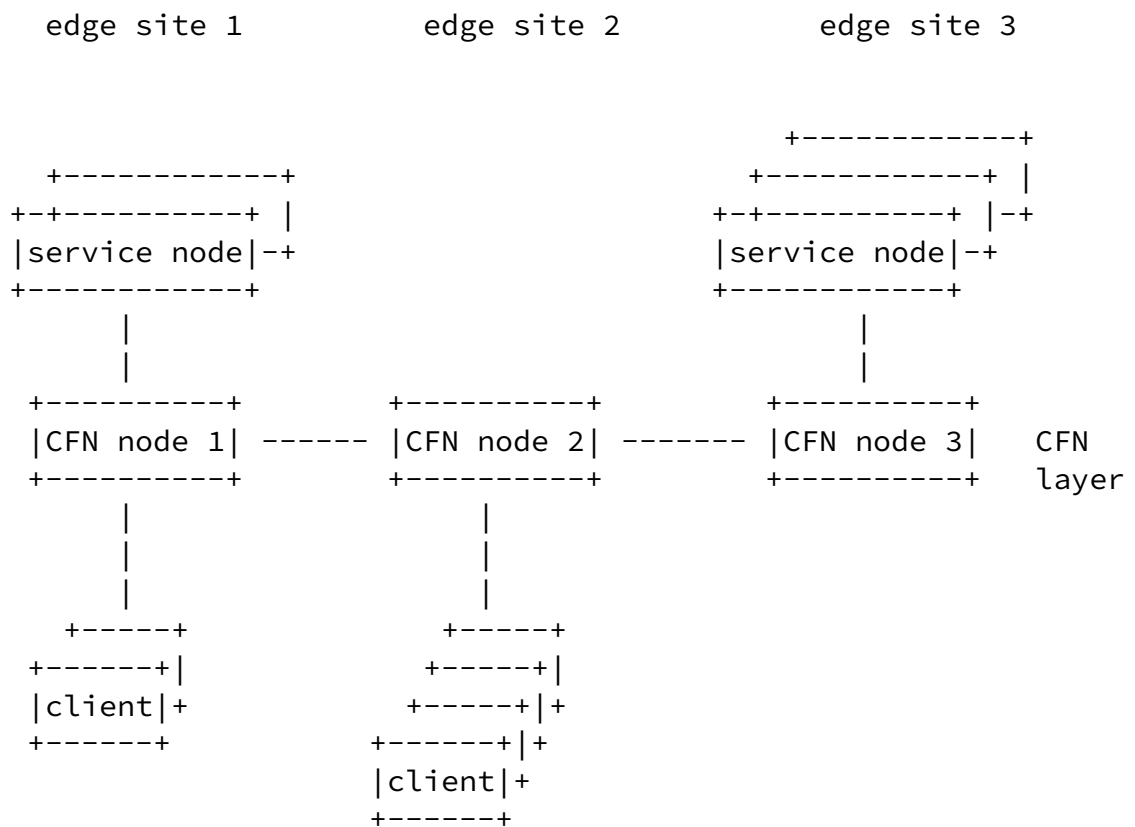


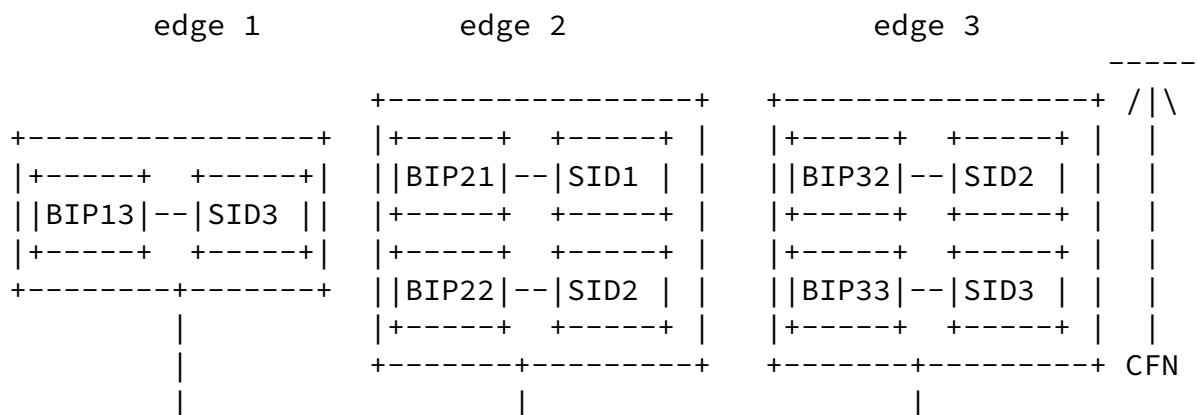
Figure 1. CFN Network Topology

2.1 CFN Service Overview

CFN uses Service ID (SID) to identify a particular service provided by service nodes on multiple edges. The end devices always use SID to initiate an access to a service. SID in current system is an anycast address. Request to a single SID can potentially be served by different edge sites. The end device does not know in advance which edge to serve the request. The procedures to make such determination is called the service dispatch. During service dispatch, the most appropriate edge site (i.e. CFN egress) is selected and it is the edge to which the service node that handles this specific request is attached to. A binding IP (BIP) address to the requested SID is known by CFN egress. BIP is a unicast IP address accessible to a particular service node providing service SID.

As shown in figure 2, service with SID 2 can be served by either CFN node 2 with binding IP BIP22 or CFN node 3 with BIP32. When the service request from the end device to SID2 reaches the ingress CFN (which is CFN node 1 in this case), the ingress CFN node should determine on the fly which egress CFN this request should be sent to. Then, the de facto service node is determined, and all the subsequent

data packets from the same flow to access this service should always be sent to the binding IP of the selected service node.



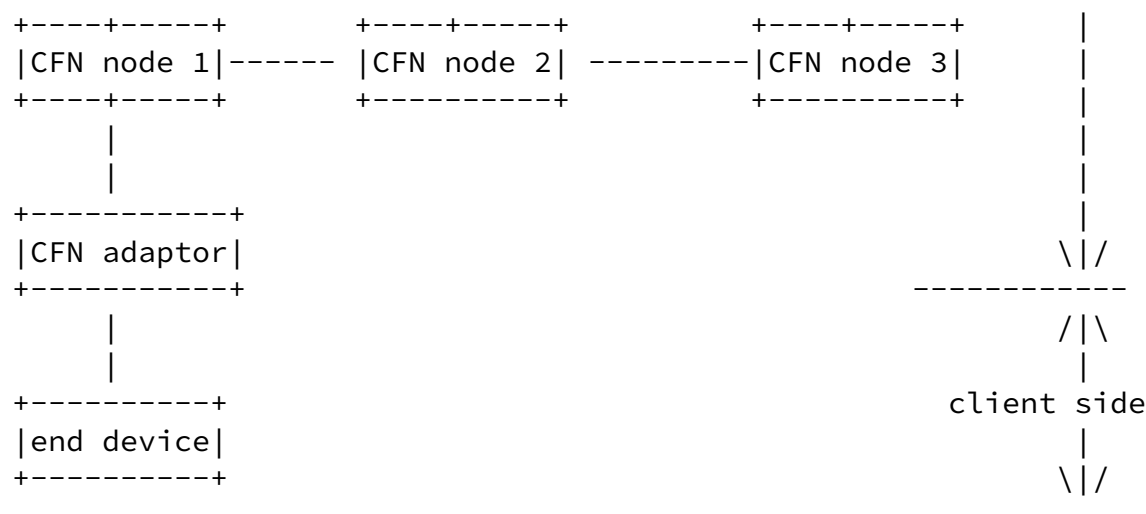


Figure 2. CFN System Overview

CFN adaptor shown in figure 2 is an entity to help the end device working with CFN in a way of keeping the binding information, identifying the initial request packet, and so on. It can be implemented as a part of CFN node (internal mode) or on a separate equipment (external mode). Figure 2 shows an external mode of CFN adaptor which can be deployed at the client side, on a virtual gateway connecting multi user equipments (UEs), as a user Plane Function (UPF) in the mobile network, or on Broadband Remote Access Server (BRAS) in the fixed network. The reason to have such an external mode is that CFN adaptor can be put closer to the clients, and then CFN node is put at some aggregated point with multiple CFN adaptors attached to it. Compared to the internal mode, external CFN adaptor keeps less binding information of the clients. It results in

less memory requirements on CFN node. CFN adaptor has no control plane.

2.2 Generic Workflow

The following procedures describe how CFN works in general.

- 1) CFN adaptor identifies a new service request from the end device, possibly by the special anycast address range for a SID.
- 2) CFN adaptor sends the request to its attaching CFN node which is CFN ingress.
- 3) CFN ingress determines the most appropriate CFN egress based on the computing resource consumption of the service nodes, the network status to the egress nodes and other information. CFN ingress forwards the request to the selected CFN egress. CFN ingress can select itself to serve the request. In this case, it is both ingress and egress in concept.
- 4) CFN egress receives the request from the CFN ingress and explicitly uses the binding IP BIP as destination address to access the required service.
- 5) CFN adaptor of ingress keeps the binding information on (SID, CFN egress) for the flow.
- 6) CFN ingress sends the subsequent packets for the same service from the same flow to the bound CFN egress to ensure the flow affinity.
- 7) CFN nodes distribute the service nodes status like available computing resources for specific services to each other on a regular base.

3. Control Plane and Data Plane

3.1 Control plane

CFN node needs to notify each other about service IDs (SIDs) attaching to it and the computing load information available corresponding to each service ID. This is used for service discovery and dispatch when a request to access a SID is received. Such information can be carried in current BGP [[RFC4760](#)] /IGP routing protocol extension. The network cost to a CFN node can be distributed in the same way. A sample service status information to be stored on

a CFN edge is shown in figure 2.

Destination	Computing Load	Network Cost	Next Hop
SID 1	3	5	CFN Egress node 1

Figure 2. Example of service status information in CFN

Computing load can be calculated from different weighted dimensions, e.g. CPU used, number of session being served, query per second, computation delay and so on. Such information needs to be refreshed regularly. In order to avoid fluctuation, it is distributed only when the metrics variation exceeds a threshold or the updating timer is expired. At the same time, the most appropriate egress node selected by the CFN ingress does not necessarily mean the one with the lowest load. Request can be sent to one selected from those egresses with relatively low computing load to avoid fluctuation.

Since SID is an anycast address, CFN ingress determines which CFN egress to forward the request to a specific SID to based on a combination of computing load and network cost.

Figure 3 shows how CFN control plane works in general. It depicts that CFN node 3 distributes computing information for service SID2. CFN node 2 should distribute service SID 2 information in the similar way as shown in figure 3. Definition and operations to extend control plane routing protocol to support CFN information distribution, and schemes/criteria to select CFN egress with anycast address from those information are to be added.

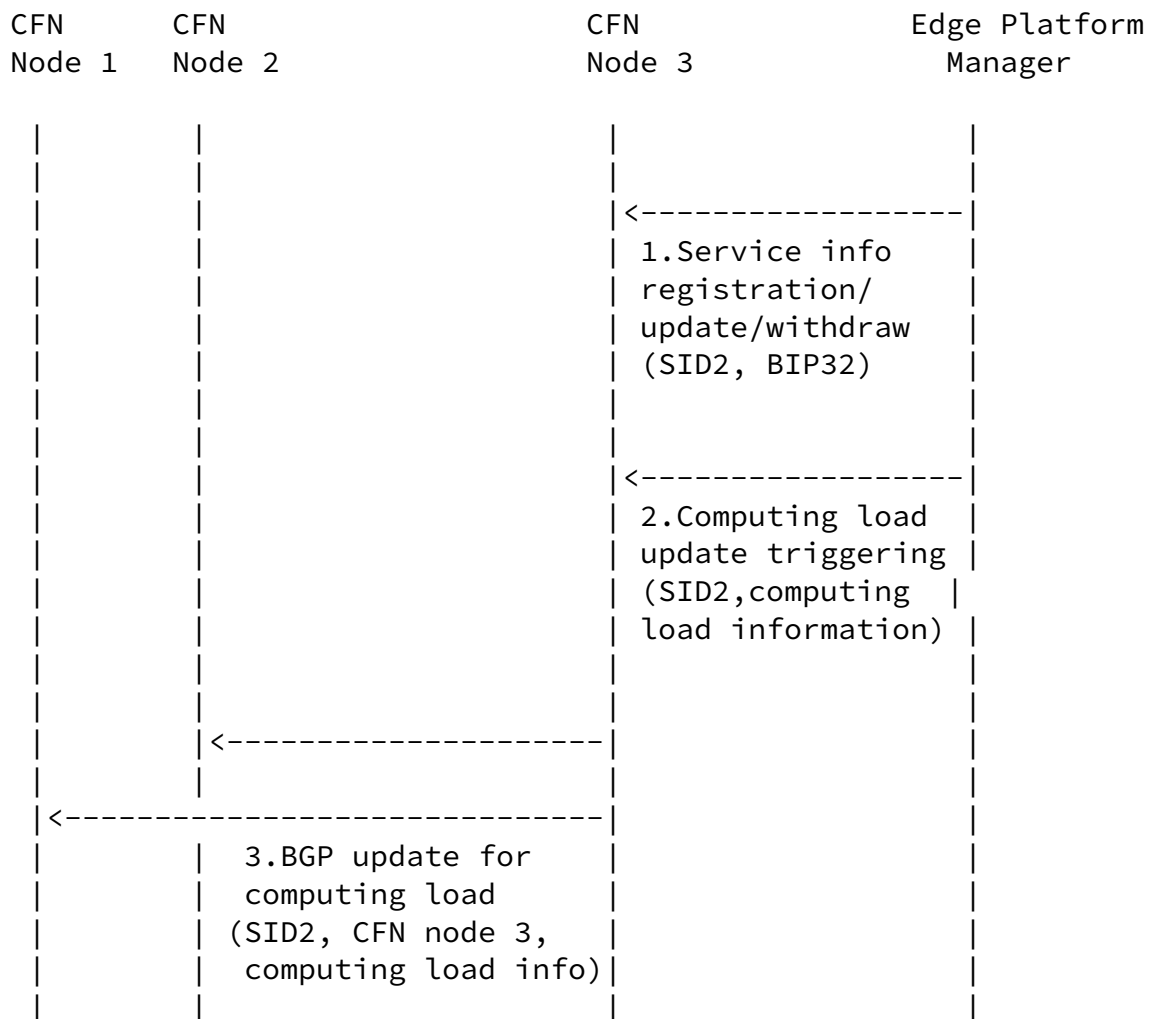


Figure 3. CFN control plane

[3.2](#) Data plane

The traditional anycast is normally used for single request single response style communication as different requests may be sent to different places when the network status changes. CFN used in edge computing may require multiple request multiple response style communication between the end device and the service node. Therefore the data plane must maintain flow affinity to ensure that the requests from the same flow are always processed by the same edge and that edge is determined at the time when the first anycast request is received by CFN ingress. The service access to the same SID from different end hosts attaching to the same CFN ingress may be dispatched to different CFN egresses. We call such a feature dynamic

anycast or Dyncast in this document.

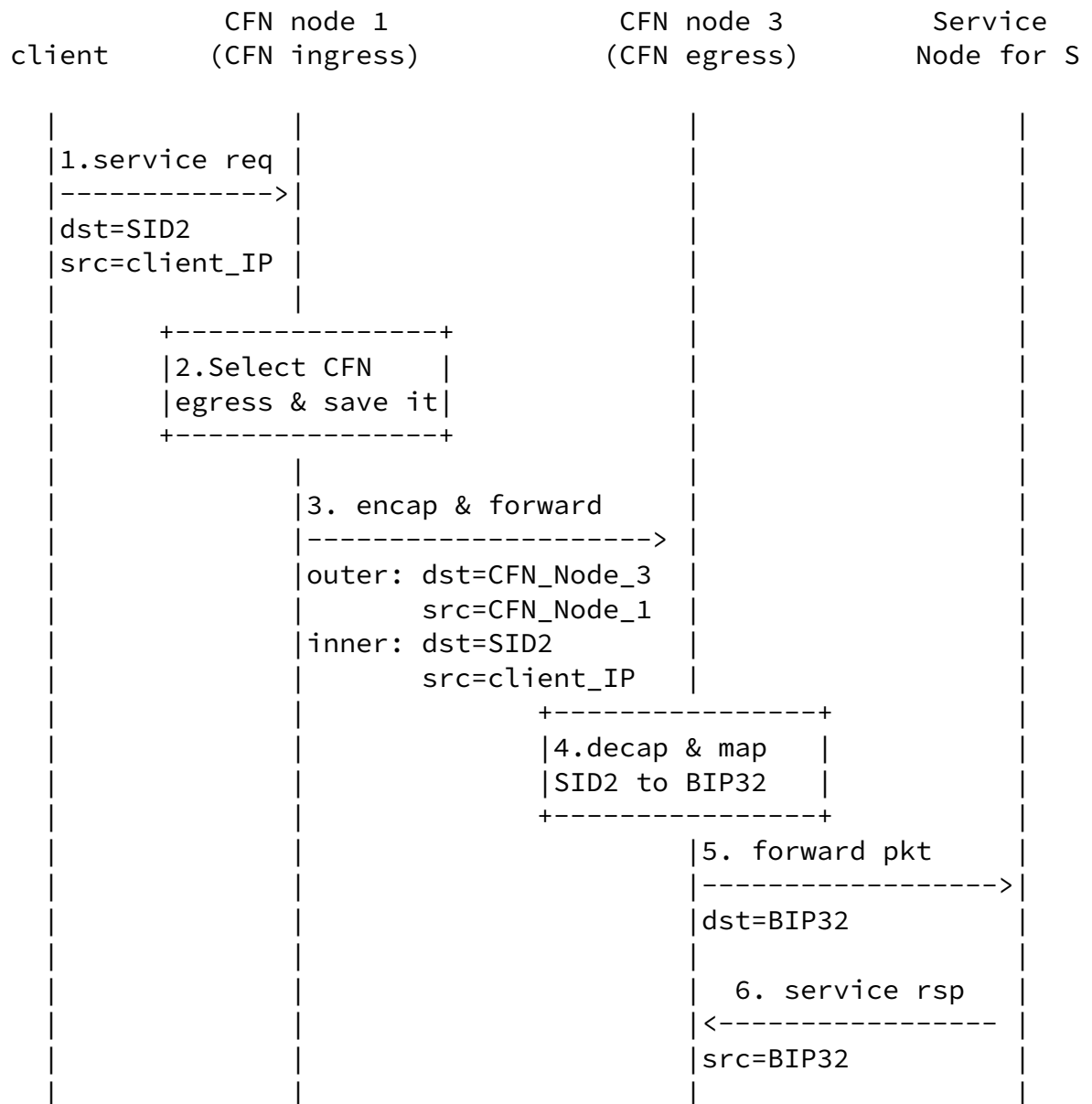
Dyncast puts some requirements on the data plane. The flow affinity table needs to be maintained by CFN ingress. On the other hand, large number of end hosts may attach to a CFN node. Therefore CFN ingress may require large memory space, such as tens of thousands of entries, to maintain such a big table of (flow, service ID, egress CFN). It is preferable to place such a binding table on an external CFN adaptor as CFN adaptor only needs to maintain a much smaller table, usually less than a hundred.

Figure 4 shows how CFN data plane works in general.

INTERNET DRAFT

Framework of CFN

Nov 2019



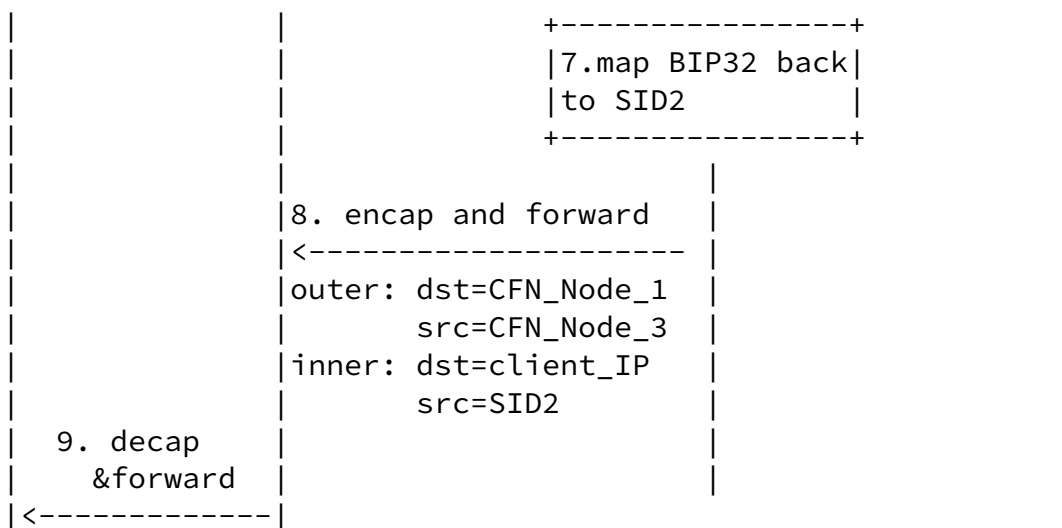


Figure 4. CFN data plane for the first request of a flow

The data plane supports the following functions.

- CFN ingress forwards the first service access request packet of a flow to the selected CFN egress by encapsulation, source routing or segment routing. Figure 4 shows the example of forwarding by encapsulation.
- CFN ingress can inform the external CFN adaptor (if there is) about the binding information on (flow, service ID, egress CFN).
- CFN adaptor (internal or external to CFN ingress) maintains the binding information table for all end hosts attaching to it and forwards the subsequent packets based on the binding information if any.

4. Summary

This draft introduces a CFN framework that enables the service request to be sent to an optimal edge to improve the overall system load balancing. It can dynamically adapt to the computing resources consumption and network status on edges and avoid overloading a single load. CFN is a network based solution that supports a large number of edges and is independent of the applications or services

hosted on the edge.

This present document is a strawman for defining CFN framework. A routing protocol (BGP [[RFC4760](#)]/IGP based) extension to distribute computing resource information and a late binding based dynamic anycast are to be defined on control plane and data plane respectively.

[5.](#) Security Considerations

The computing resource information changes over time very fast with the creation and termination of service instance handlers. When such information is carried in routing protocol, too many updates can make the network fluctuate. [Section 3.1](#) gives a brief idea on avoiding sending too much updates.

[6.](#) IANA Considerations

No IANA action is required so far.

[7.](#) Acknowledgements

Li, et al

[Page 12]

INTERNET DRAFT

Framework of CFN

Nov 2019

[8.](#) References

[8.1](#) Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

[8.2](#) Informative References

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", [RFC 4760](#), January 2007.

[CFN-req] Geng, L., et al, "Compute First Networking (CFN) Scenarios and Requirements", [draft-geng-rtgwg-CFN-req-00](#), November 2019.

Authors' Addresses

Yizhou Li
Huawei Technologies

Email: liyizhou@huawei.com

Jeffrey He
Huawei Technologies

Email: jeffrey.he@huawei.com

Liang Geng
China Mobile
Email: gengliang@chinamobile.com

Peng Liu
China Mobile
Email: liupengyjy@chinamobile.com

Yong Cui
Tsinghua University

Email: cuiyong@tsinghua.edu.cn

