

Workgroup:

Application-Layer Traffic Optimization

Internet-Draft: draft-liu-alto-can-usecase-01

Published: 20 January 2022

Intended Status: Informational

Expires: 24 July 2022

Authors: P. Liu

Y. Fu

China Mobile

China Mobile

## **Computing-aware Networking Use case of ALTO**

### **Abstract**

The ever-emerging new services are imposing more and more highly demanding requirements on the network. In order to meet these requirements, some new technology trends of network emerge as the times require. On the one hand, for the selection of service node and forwarding path, in addition to considering the network topology and link state, more factors are also considered, such as the computing properties of service node; On the other hand, network and application present the trend of mutual perception, including application to perceive the state of network path, or network to perceive the demand of application.

This draft describes a new network scenario and architecture considering computational properties, and assumes that Alto could be used as an important node to realize the deployment of services, and to assist in the selection of service nodes.

### **Requirements Language**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

### **Status of This Memo**

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on 24 July 2022.

## Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

- [1. Introduction](#)
- [2. Usage Scenarios of CAN](#)
  - [2.1. AR/VR](#)
  - [2.2. Internet of Vehicles](#)
- [3. CAN Framework and ALTO](#)
- [4. Deployment of CAN with ALTO](#)
- [5. Scheduling of CAN with ALTO](#)
- [6. Summary](#)
- [7. Security Considerations](#)
- [8. IANA Considerations](#)
- [9. Acknowledgements](#)
- [10. Informative References](#)
- [Authors' Addresses](#)

## 1. Introduction

For new services with heavy computing tasks, such as AR/VR, video recognition and so on, the computing time and network transmission delay are almost the same order of magnitude. In this kind of scenario, computing attributes become more important than traditional services.

The generation of edge computing is to solve such problems. Edge computing is to deploy service nodes close to the user side, which shortens the distance of network transmission. Moreover, it can deploy specific computing resources, such as CPU/GPU, to meet the needs of different services.

It is predicted by Gartner that by 2025, more than 50% of the computing data needs to be analyzed, processed, and stored at the edge. Since the service providers begins to offer the edge computing

infrastructure to provide better response time and transfer rate. There are also some challenges of edge computing itself, which are pointed out in the work of dyncast [draft-liu-dyncast-ps-usecases-01], [draft-li-dyncast-architecture-00] :

- \* Geographically Scattered Large Number of Edge Sites. The edge sites are highly distributed and may not have proximate distances to user.
- \* Resource Limitation. There are fewer servers of server per node.
- \* Heterogeneous Hardware, such as CPU, GPU, Memory, ASICs.
- \* Dynamic Load. The available resources may change quickly.
- \* Edge-cloud Coordination. Edge does not solve all requests.
- \* High Cost. On-site maintenance is expensive.
- \* Mission Critical. Users are counting on you for 100% reliability of industry automation.

So how to collaboratively deploy and computing services based on the computing resources in network to meet the diverse computing requirements, and achieve the on-demand allocation and dispatch of service request needs be studied.

Some existing works have begun to consider the computing attributes. For example, coinrg initiated the consideration of computing and storage resources. Dyncast [[I-D.liu-dyncast-ps-usecases](#)][[I-D.li-dyncast-architecture](#)] proposed how to introduce the scheme of computing metric at the routing level. Semantic routing[], which also extends the semantics of routing in a broad sense. However, today's routing system and technology has been relatively good, the introduction of more metric in routing still need more theoretical and experimental verification. In the work of ITU, it is more from the perspective of architecture, such as ITU Y.CAN-reqts [Y.CAN-reqts: "functional requirements of Computing-aware Networking"] proposed a new network architecture-computing-aware networking (CAN), CAN schedules service request to the optimal computing site along optimal path to meet service requirements both on network and computing. ITU.Y.CPN-arch [Y.CPN-arch: "Framework and architecture of Computing power Network"] provides the framework and architecture of Computing power Network, specifies its functional entities and defines the functionalities of these functional entities. So the convergence of network and computing brought by edge computing includes the issue of service deployment and service request scheduling. ALTO has done the work of collect the network information for application, it may help to do some work in the two important issues:

How to deploy service nodes based on computing resources. For this point, [draft-contras-alto-service-edge-02] gives the corresponding idea of using Alto to deploy edge computing nodes. Alto can better interact with the upper application, fully understand the requirements of the application, including computing requirements and collect the information of infrastructure resources.

How to select the most suitable node for service request. Alto can also help this kind of work to a certain extent. Centralized selection of service nodes and paths is relatively easy to implement, such as SDN. However, emerging service requests require high real-time performance, and there may be some efficiency and complexity problems, which have been analyzed in the work of dyncast.

## 2. Usage Scenarios of CAN

Any multi-point deployment service that requires high computing power or low latency will involve the joint scheduling of network and computing resources.

### 2.1. AR/VR

Take the AR/VR as an example. The upper bound latency for motion-to-photon (MTP) is less than 20 milliseconds to avoid the motion sickness. It consists of four parts, and the frame rendering computing delay is 5.5 milliseconds, so the network delay demand can be calculated as 5.1milliseconds.

+-----+-----+		
Total MTP delay	50ms	
+-----+-----+		
sensor sampling delay	<1.5ms	
+-----+-----+		
display refresh delay	5 ms	
+-----+-----+		
frame rendering delay	5.5ms	
computing with GPU		
+-----+-----+		
network delay	$20-1.5-5.5-7.9=5.1\text{ms}$	
+-----+-----+		

Figure 1: Delay of MTP

So the budgets for computing delay and network delay are almost equivalent. Considering another factor that the computing resources have a big difference in different edges, it is necessary to apply dynamically steer traffic to the 'best' edge.

## 2.2. Internet of Vehicles

Under the scenario of Internet of Vehicles, the services are divided into auxiliary driving and on-board entertainment services . For the auxiliary driving function, for road traffic conditions outside the line of sight due to obstructions, blind areas, etc., the edge computing node obtains comprehensive road traffic information around the location of the vehicle, performs unified data processing, and sends out warning signals for vehicles with potential safety hazards, to assist the safe driving of vehicles.

Apparently, there are obviously differences between services requirements of auxiliary driving services and entertainment services, which needs to be processed by different edge nodes

## 3. CAN Framework and ALTO

In order to realize ubiquitous computing and service awareness, interconnection and collaborative scheduling, the computing-aware networking architecture can be divided into computing service layer, computing resource layer, computing routing layer, network resource layer, and computing and network management layer. Among them, the computing routing layer contains the control plane and data plane which is shown in Figure 2. Based on the ubiquitous computing resources of the network, the computing resource layer abstracts and models based on a unified measurement and modeling template, and announces computing information to the computing routing layer. And then the computing routing layer comprehensively considers user needs and network resource status and computing resource status, to schedule service requests to appropriate computing nodes. In addition, the computing management layer completes the control and management of computing resources.

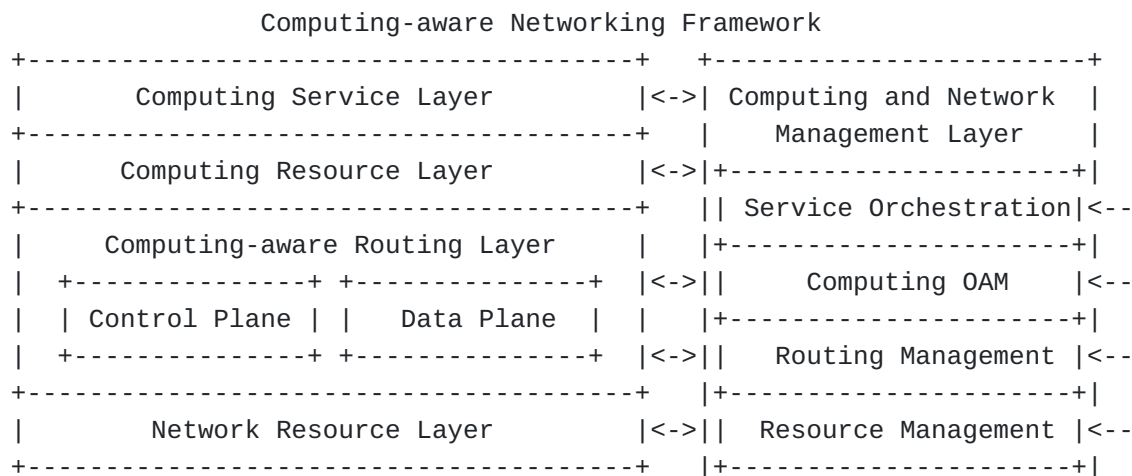


Figure 2: CAN Framework and ALTO

\* Computing service layer: computing service layer is computing service provider, which deploys, operates and manages many kinds of computing services and applications. In addition, it can realize the functions of service decomposition and service scheduling through the API gateway.

\* Computing resource layer: it is based on the existing computing infrastructure, and includes a combination of computing resources from single-core CPU to multi-core CPU, CPU+GPU+FPGA, etc. And it could supply computing modeling function, computing API function, computing resource identification and other functions to meet the diverse computing needs of different applications based on physical computing resources.

\* Computing-aware routing layer: It contains control plane and data plane, performs computing-aware routing and generates service scheduling policy in network layer. Based on the discovery of abstracted computing and network resources, computing routing layer generates new routing tables which include the information of computing in network, considers the state of network and computing comprehensively, and thus generates routing policy for different service requests. Network resource layer: It utilizes the existing network infrastructure, which includes access network, metropolitan area network and backbone network, to provide ubiquitous network connection.

\* Computing and network management layer: It adds management functions towards computing resources and computing services based on the traditional network management function. Therefore, the computing and network management layer performs service orchestration, resource management, routing measurement and computing OAM. In addition, the computing and network management layer could be used to perform the pre-configuration function and management function, which interacts with each functional layer.

\* Network resource layer: using the existing network infrastructure to provide network connection, network infrastructure includes access network, metropolitan area network and backbone network.

Based on the five functional modules defined above, computing-aware networking can realize the awareness, control and scheduling of computing and network resources, and further perform dynamic and on-demand service scheduling. The function of computing and network management layer may be realized by Alto or by opening interface with Alto. Considering a single ALTO Client as part of the Computing and Network Management Layer aggregating all the requests towards the ALTO Server, this also could decouple the solution from a specific CAN architecture.

#### 4. Deployment of CAN with ALTO

With the development of edge computing, there is multiple services duplication deployed in different edge nodes. To improve the effectiveness of service deployment, the problem of how to choose optimal edge node to deploy services needs to be solved. More stable static information should be considered in service deployment, [[I-D.contreras-alto-service-edge](#)] introduces the consideration of depoly applications or functions to the edge, such as the type of instance, interface option associated bandwidth of the network interface, compute flavor of CPU/GPU, etc, optional storage extension, optional hardware acceleration characteristics.

Besides those, more network and service factors may to be considered to meet the CAN Framework, such as

- \* Network and computing resource topology: the overall consideration of network access, connectivity, path protection or redundancy. and the location and overall distribution of computing resources in network, and the relative position towards network topology.

- \* Location: the number of users brought, the differentiation of service types and number of connections requested by users, etc. For edge nodes located in popular area, which with large amount of users and service requests, the service duplication can be deployed more than other areas.

- \* Capacity of multiple edge nodes: not only a single node, but also the total number of requests that can be processed by the resource pool composed of multiple nodes

- \* Service category: For example, whether the business is multi-user interaction, such as video conferencing, games, or just resource acquisition, such as short video viewing Alto can help to obtain one or more of the above information, so as to provide suggestions or formulate principles and strategies for service deployment.

For the collection of those information, seconds level or minutes level frequency is enough, while serious real-time processing isn't necessary. For example, periodically collecting the total consumption of computing resources, or the total number of sessions accessed, to notify where to depoly more VMS or containers. Unlike the scheduling of request, service deployment should still follow the principle of proximity. The more local access, the more resources should be deployed. If the resources are insufficient, the operator can be informed to increase the hardware resources. Alto can be used to collect and reprot thoes information.

## 5. Scheduling of CAN with ALTO

Compared to the deployment, scheduling needs to consider more dynamic information to select and adjust the optimal (rather than the shortest) path in real time, such as:

- \* Mobility: CAN schedules service request to the optimal service node among several service nodes near to users. So when user mobiles, the nearby service nodes changes and new scheduling are needed to choose new path and new service node.
- \* Real time delay of network: network delay is always in the process of dynamic change, and more and more services propose strict time requirements, which is one of the most important factors affecting user experience.
- \* Real time status of computing resources: computing resources change frequently and the status of computing resources heavily affect the completion time of service, which is also one of the most important factors affecting user experience.
- \* Comprehensive status of network status and computing status: the update frequency of computing status and network status is different, it is necessary to generate a comprehensive value to reflect the status of them. Collecting the information from multi-protocols will bring the issue of synchronization, which is not easy and cause some additional expenses. (If the network is deterministic network that support synchronization such as detnet, it will be better.) One protocol may be the right way.
- \* Various service requirements: different services propose different service requirements on computing and network, including bandwidth, latency, computing resources etc, and the latency includes both transmission latency in network and processing latency in service node, for transmission-intensive services, the transmission latency accounts a lot, so the network latency of services are more important.
- \* Availability of network or computing resources: such as temporary unavailability caused by network equipment or service node failure.

Alto can still help collect network and service node information,

- \* Providing the best choice of network and service nodes. Based on the collected network information, computing information, and service requirements on network and computing, Of course, there are still some real-time and complexity problems.
- \* Providing data analysis and policy distribution, real-time node selection still depends on distributed routing, such as dyncast.



[[I-D.contreras-alto-service-edge](#)] introduces how to use BGP to realize it. BGP is an option where [[RFC7971](#)] shows BGP prefixes, AS numbers, AS distances, or other BGP metrics could be collected. However, The ALTO service may not know the instantaneous congestion status of the network, all link bandwidths, all information about the actual routing and whether the candidate endpoint itself is overloaded. So it may not be satisfied for the application with very real time requirements. "real-time" could be relative and may affect the result of scheduling. If it is good, the scheduling can be more accurate and better meet the application requirements; If the real-time performance is not good, the network or node state may change when the flows arrive, resulting in the demand can not be met. So how to improve the performance of BGP or announce the corresponding information through other ways need to be research.

## 6. Summary

The converge of network and computing, as well as the interaction with applications has become one of the current technical development directions. This draft analyzes the development status of the current computing aware network and the functional modules in its architecture that can interact with Alto. As a protocol to connect networks and applications, Alto may play a more important role.

## 7. Security Considerations

TBD.

## 8. IANA Considerations

TBD.

## 9. Acknowledgements

Thanks to Yizhou Li, Qin Wu, Tianji Jiang for helpful suggestion. Thanks go to Dirk Trossen, Luigi Iannone, and Carsten Bormann for their inspiring Dyncast work.

## 10. Informative References

[**I-D.contreras-alto-service-edge**] Contreras, L. M., Lachos, D. A., Rothenberg, C. E., and S. Randriamasy, "Use of ALTO for Determining Service Edge", Work in Progress, Internet-Draft, draft-contreras-alto-service-edge-03, 12 July 2021, <<https://www.ietf.org/archive/id/draft-contreras-alto-service-edge-03.txt>>.

[**I-D.li-dyncast-architecture**] Li, Y., Iannone, L., Trossen, D., and P. Liu, "Dynamic-Anycast Architecture", Work in Progress,

Internet-Draft, draft-li-dyncast-architecture-00, 15 February 2021, <<https://www.ietf.org/archive/id/draft-li-dyncast-architecture-00.txt>>.

[I-D.liu-dyncast-ps-usecases] Liu, P., Willis, P., and D. Trossen, "Dynamic-Anycast (Dyncast) Use Cases & Problem Statement", Work in Progress, Internet-Draft, draft-liu-dyncast-ps-usecases-01, 15 February 2021, <<https://www.ietf.org/archive/id/draft-liu-dyncast-ps-usecases-01.txt>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC7971] Stiemerling, M., Kiesel, S., Scharf, M., Seidel, H., and S. Previdi, "Application-Layer Traffic Optimization (ALTO) Deployment Considerations", RFC 7971, DOI 10.17487/RFC7971, October 2016, <<https://www.rfc-editor.org/info/rfc7971>>.

#### Authors' Addresses

Peng Liu  
China Mobile  
Beijing  
100053  
China

Email: [liupenggyjy@chinamobile.com](mailto:liupenggyjy@chinamobile.com)

Yuexia Fu  
China Mobile  
Beijing  
100053  
China

Email: [fuyuexia@chinamobile.com](mailto:fuyuexia@chinamobile.com)