Workgroup: rtgwg
Internet-Draft: draft-liu-dyncast-reqs-02
Published: 7 March 2022
Intended Status: Informational
Expires: 8 September 2022
Authors: P. Liu T. Jiang P. Eardley
China Mobile China Mobile British Telecom
D. Trossen C. Li
Huawei Technologies Huawei Technologies

Dynamic-Anycast (Dyncast) Requirements

Abstract

This draft provides requirements for an architecture addressing the problems outlined in the use case and problem statement draft for Dyncast[I-D.liu-dyncast-ps-usecases] .

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Computing service instances instantiated at multiple geographical edge sites are used to better realize an edge computing service in edge computing use cases, as shown in[I-D.liu-dyncast-ps-usecases]. To optimally deliver the service request to the most appropriate service instance is the fundamental requirement in such deployments. As shown in [I-D.liu-dyncast-ps-usecases], choosing the most appropriate service instance should take both, the computing resources available and the network path quality, into consideration. "Optimal" here additionally means the architecture and overall mechanism should be efficient, support high dynamism, while maintaining instance affinity, as shown in [I-D.liu-dyncast-ps-usecases].

This draft provides the requirements to realize the potential dynamic anycast architecture by alleviating the problems of existing solutions outlined in [I-D.liu-dyncast-ps-usecases]

2. Definition of Terms

Service: A monolithic functionality that is provided by an endpoint according to the specification for said service. A composite service can be built by orchestrating monolithic services.

Service instance: Running environment (e.g., a node) that makes the functionality of a service available. One service can have several instances running at different network locations.

Service identifier: Used to uniquely identify a service, at the same time identifying the whole set of service instances that each represent the same service behaviour, no matter where those service instances are running.

Anycast: An addressing and packet sending methodology that assign an "anycast" identifier for one or more service instances to which requests to an "anycast" identifier could be routed, following the definition in [RFC4786] as anycast being "the practice of making a particular Service Address available in multiple, discrete, autonomous locations, such that datagrams sent are routed to one of several available locations".

Dyncast: Dynamic Anycast, taking the dynamic nature of computing resource metrics into account to steer an anycast-like decision in sending an incoming service request.

3. Desirable System Characteristics and Requirements

In the following, we outline the desirable characteristics of a system to overcome the observed problems in [I-D.liu-dyncast-ps-usecases] for the realization of the use cases described in that document.

3.1. Anycast-based Service Addressing Methodology

A unique service identifier is used by all the service instances for a specific service no matter which edge it attaches to. An anycast like addressing and routing methodology among multiple edges makes sure the data packet can potentially reach any of the edges with the service instance attached. At the same time, each service instance has its own unicast address to be used by the attaching edge to access the service.Since a client will use the service identifier as the destination addressing, mapping of the service identifier to the unicast address will need to happen in-band, considering the metrics for selection to make this selection service-specific. From an addressing perspective, a desirable system for the realization of the use cases described in that document.

o MUST provide a discovery and mapping methodology for the in-band mapping of the service identifier (an anycast address) to a specific unicast address.

3.2. Instance Affinity

A routing relation between a client and a service exists not at the packet but at the service request level in the sense that one or more service requests, possibly consisting of one or many more routing-level packets, must be ensured to be sent to said service.Each service may be provided by one or more service instances, each providing equivalent service functionality to their respective clients, while those service instances may be deployed at different locations in the network. With that, the routing problem becomes one between the client and a selected service instance for at least the duration of the service-level request, but possibly more than just one request.

This relationship between the client and the chosen service instance is described as "instance affinity" in the following, where the "affinity" spans across the aforementioned one or more service requests. This impacts the routing decision to be taken in that the normal packet level communication, i.e., each packet is forwarded individually based on the forwarding table at the time, will need extending with the notion of instance affinity since otherwise individual packets may be sent to different places when the network status changes, possibly segmenting individual requests and breaking service-level semantics.

The nature of this affinity is highly dependent on the nature of the specific service. The minimal affinity of a single request represents a stateless service, where each service request may be responded to without any state being held at the service instance for fulfilling the request. Providing any necessary information/ state in-band as part of the service request, e.g., in the form of a multi-form body in an HTTP request or through the URL provided as part of the request, is one way to achieve such stateless nature. Alternatively, the affinity to a particular service instance may span more than one request, as in our VR example in [I-D.liu-dyncast-ps-usecases], where previous client input is needed to render subsequent frames. Therefore, a desirable system

o MUST maintain "instance affinity" which MAY span one or more service requests, i.e., all the packets from the same flow MUST go to the same service instance.

3.3. Proper Runtime-state Granularity and Keeping

The instance affinity, as outlined in Section 3.2, requires a client and the chosen service instance to keep persistent relationship across one or more service requests. For a multi-request session, this determines that the mapping logic has to consistently pick up the same service instance. This type of affinity can be normally achieved by deploying a mapping device to keep in-place all the necessary states. However, a client, e.g., a mobile UE, has generally many applications running. If all, or majority, of the applications request the dyncast-like services, then the runtime states that need to be created and accordingly maintained would require high granularity. In the extreme scenario, this granular requirement could reach the level of per-UE per-APP per-(sub)flow with regard to a service instance.

Evidently, these fine-granular runtime states can potentially become heavy burden for network devices if they have to dynamically create and maintain them. On the other hand, it is not appropriate either to place the state-keeping task on clients themselves. Therefore, a desirable system

o MUST avoid keeping fine runtime-state granularity in network nodes in order to achieve instance affinity.

o MUST provide mechanism to free clients from maintaining granular runtime-states in order to achieve instance affinity.

3.4. Encoding Metrics

As outlined in the scenarios in [I-D.liu-dyncast-ps-usecases], metrics can have many different semantics, particularly if considered to be service- specific. Even the notion of a "computing load" metric may be computed in many different ways. What is crucial, however, is the representation and encoding of that metric when being conveyed to the routing fabric in order for the routing elements to act upon those metrics. Such representation may entail information on the semantics of the metric or it may be purely one or more semantic-free numerals. Agreement of the chosen representation among all service and network elements participating in the service-specific routing decision is important. Specifically, a desirable system

o MUST agree on the service-specific metrics and their representation between service elements in the participating edges in the network and network elements acting upon them.

o MAY obfuscate the specific semantic of the metric to preserve privacy of the service provider information towards the network provider.

o MAY include routing protocol metrics

3.5. Signaling Metrics

The aforementioned representation of metrics needs conveyance to the network elements that will need to act upon them. Depending on the service-specific decision logic, one or more metrics will need to be conveyed. Problems to be addressed here may be that of loop avoidance of any advertisement of metrics as well as the frequency of such conveyance and therefore the overall load that the signaling may add to the overall network traffic. While existing routing protocols may serve as a baseline for signaling metrics, other means to convey the metrics can equally be realized. Specifically, a desirable system

o MUST provide mechanisms to signal the metrics for using in routing decisions

o MUST realize means for rate control for signaling of metrics

o MUST implement mechanisms for loop avoidance in signaling metrics, when necessary

3.6. Using Metrics in Routing Decisions

Metrics being conveyed, as outlined in Section 3.4, in the agreed manner, as outlined in Section 3.3, will ultimately need suitable action in the routers of the network. Routing decisions can be manifold, possibly including (i) min or max over all metrics, (ii) extending previous action with a random or first choice when more than one min/max entry found, (iii) weighted round robin of all entries, among others. It is important for the proper work of the service-specific routing decision, that it is understood to both network and service provider, which action (out of a possible set of supported actions) is to be used for a particular set of metrics. Specifically, a desirable system

Further, different network nodes, e.g., routers, switches, etc., bear diversified capabilities even in the same routing domain, let alone in different administrative domains. So, the service-specific metrics that have been adopted by some nodes might not be supported by others, either due to technical reasons, administrative reasons, or something else. There could be some scenario that a node supporting service-specific metrics might prefer some type of metrics to others [3GPP-TR22.847], or, in another scenario, even not utilize any at all. Therefore, there must exist flexibility in term of metrics handling and routing decisions in a network.

o MUST specify a default action to be taken, if more than one action possible o MUST allow a network node not supporting service-specific metrics to interoperate with the supporting ones, i.e., providing backward compatibility.

o SHOULD allow the prioritization of using the service-specific metrics when compared to the currently widely-used networking metrics, like bandwidth, delay, loss, etc.

o SHOULD enable other alternative actions to be taken. (1)Any solution MUST provide appropriate signaling of the desired action to the router. For this, the action MAY be signaled in combination with signaling the metric (see Section 3.4). (2)Any solution SHOULD allow associating the desired action to a specific service identifier.

3.7. Supporting Service Dynamism

Network cost in the current routing system usually does not change very frequently. However, computing load and service-specific metrics in general can be highly dynamic, e.g., changing rapidly with the number of sessions, CPU/GPU utilization and memory space. It has to be determined at what interval or events such information needs to be distributed among edges. More frequent distribution of more accurate synchronization may result in more overhead in terms of signaling.

Choosing the least path cost is the most common rule in routing. However, the logic does not work well when routing should be aware of service-specific metrics. Choosing the least computing load may result in oscillation. The least loaded edge can quickly be flooded by the huge number of new computing demands and soon become overloaded with tidal effects possibly following.

Generally, a single instance may have very dynamic resource availability over time in order to serve service requests. This availability may be affected by computing resource capability and load, network path quality, and others. The balancing mechanisms should adapt to the service dynamism quickly and seamlessly. With this, the relationship between a single client and the set of possible service instances may possibly be very dynamic in that one request that is being dispatched to instance A may be followed by a request that is being dispatched to instance B and so on, generally within the notion of the service-specific service affinity discussed before in Section 3.2. With this in mind, a desirable system

o MUST support the dynamics of metrics changing on, e.g., a per flow basis, without violating the metrics defined in the selection of the specific service instance, while taking into account the requirements for the signaling of metrics and routing decision (see Section 3.4 and 3.5).

4. Conclusion

This document presents high-level requirements for solutions to Dyncast, where the architecture should address how to distribute the resource information and how to assure instance affinity in an anycast based service addressing environment, while realizing appropriate routing actions to satisfy the metrics provided.

5. Security Considerations

TBD

6. IANA Considerations

No IANA action is required so far.

7. Contributors

The following people have substantially contributed to this document:

Peter Willis
BT

8. Informative References

[RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, DOI 10.17487/RFC4786, December 2006, <https://www.rfc-editor.org/info/rfc4786>.

[I-D.liu-dyncast-ps-usecases] Liu, P., Willis, P., Trossen, D., and C. Li, "Dynamic-Anycast (Dyncast) Use Cases & Problem Statement", Work in Progress, Internet-Draft, draft-liu-dyncast-ps-usecases-02, 17 January 2022, <https://www.ietf.org/archive/id/draft-liu-dyncast-ps-usecases-02.txt>.

[TR22.874] 3GPP, "Study on traffic characteristics and performance requirements for AI/ML model transfer in 5GS (Release 18)", 2020.

Acknowledgements

The author would like to thank Yizhou Li, Luigi IANNONE and Geng Liang for their valuable suggestions to this document.

Authors' Addresses

Peng Liu

China Mobile

Email: liupengyjy@chinamobile.com

Tianji Jiang
China Mobile

Email: jiangtianji@chinamobile.com

Philip Eardley
British Telecom

Email: philip.eardley@bt.com

Dirk Trossen
Huawei Technologies

Email: dirk.trossen@huawei.com

Cheng Li
Huawei Technologies

Email: c.l@huawei.com