Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 7, 2020

Z. Li
S. Peng
Huawei Technologies
D. Voyer
Bell Canada
C. Xie
China Telecom
P. Liu
China Mobile
C. Liu
China Unicom
K. Ebisawa
Toyota Motor Corporation
S. Previdi
Individual
J. Guichard
Futurewei Technologies Ltd.

November 04, 2019

Application-aware IPv6 Networking (APN6) Framework
draft-li-apn6-framework-00

Abstract

A multitude of applications are carried over the network, which have varying needs for network bandwidth, latency, jitter, and packet loss, etc. Some new emerging applications (e.g. 5G) have very demanding performance requirements. However, in current networks, the network and applications are decoupled, that is, the network is not aware of the applications' requirements in a fine granularity. Therefore, it is difficult to provide truly fine-granularity traffic operations for the applications and guarantee their SLA requirements.

This document proposes a new framework, named Application-aware IPv6 Networking (APN6), which makes use of IPv6 encapsulation to convey the application characteristic information such as application identification and its network performance requirements into the network to facilitate service provisioning, perform application-level traffic steering and network resource adjustment.

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 May 7, 2020.

Table of Contents

1. Introduction

A multitude of applications are carried over the network, which have varying needs for network bandwidth, latency, jitter, and packet loss, etc. Some applications such as online gaming and live video streaming has very demanding network requirements and therefore require special treatment in the network. However, in current networks, the network and applications are decoupled, that is, the network is not aware of the applications' requirements in a fine granularity. Therefore, it is difficult to provide truly fine-granularity traffic operations for the applications and guarantee their SLA requirements accordingly.
[I-D.li-apn6-problem-statement-usecases] describes the challenges of traditional differentiated service provisioning methods, such as five tuples used for ACL/PBR causing coarse granularity, DPI imposing high CAPEX & OPEX and security issues, as well as orchestration and SDN-based solution causing long control loops.

This document proposes a new framework, named Application-aware IPv6 Networking, aiming to guarantee fine-granularity SLA requirements of applications, which make use of IPv6 encapsulation to convey the application characteristic such as application identification and its network performance requirements into the network to determine the path, steer traffic, and perform network resource adjustment.

2. Specification of Requirements

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].

This document is not a protocol specification and the key words in this document are used for clarity and emphasis of requirements language.

3. Terminology

ACL: Access Control List

APN6: Application-aware IPv6 Networking

DPI: Deep Packet Inspection

PBR: Policy Based Routing

QoE: Quality of Experience

4. APN6 Framework and Key Components

The APN6 framework is shown in Figure 1. The key components include Application-aware App, App-aware Edge Device, App-aware-process Head-End, App-aware-process Mid-Point, and App-aware-process End-Point.

Packets carry application characteristic information (i.e. application-aware information) which includes the following information:

- Application-aware identification information: identifying application, the user of application, i.e. the packets as part of the traffic flow belonging to a specific SLA level/Application/ User;

- Network performance requirements information: specifying at least one of the following parameters: bandwidth, delay, delay variation, packet loss ratio, security, etc.

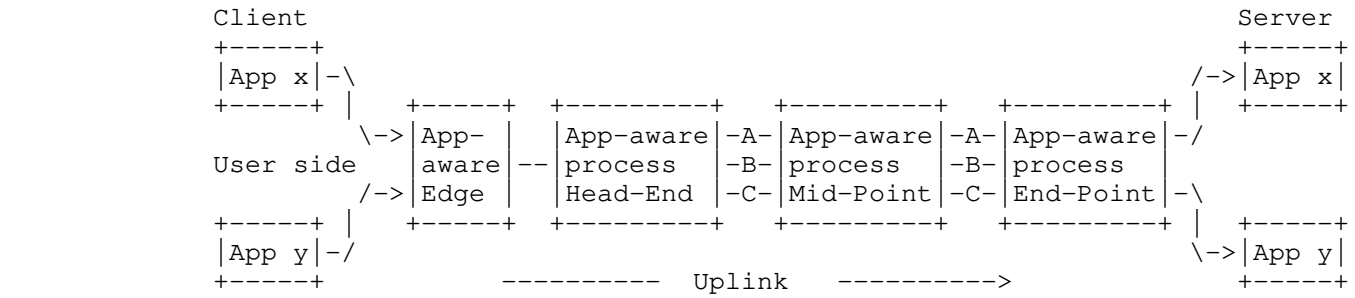

Figure 1 APN6 Framework and Key Components

The key components are introduced as follows.

1. Application-aware App: The host obtains the application characteristic information of the Application-aware App and generates the packets which carry the application characteristic information in IPv6 encapsulation. If carried in the packets, this information is used by the App-aware-process Head-End to determine the path between the App-aware-process Head-End and the App-aware-process End-Point for forwarding the packets to their destination, that is, to steer the packet in to a given policy which satisfies the application requirements. In the APN6 framework, the application is not mandatory to be application-aware.

2. App-aware Edge Device: This network device receives packets from applications and obtains the application characteristic information. If the application is not Application-aware App, the application

characteristic information can be obtained by packet inspection, derived from services information such as double VLAN tagging (C-VLAN and S-VLAN), or added according to the local policies which is out of the scope of this document. The App-aware Edge Device adds the application characteristic information in IPv6 encapsulation on behalf of the application. The packets carrying the application characteristic information will be sent to the App-aware-process Head-End, and the application characteristic information will be used to determine the path between the App-aware-process Head-End and the App-aware-process End-Point for forwarding the packets.

3. App-aware-process Head-End: This network device receives packets and obtains the application characteristic information. A set of paths, tunnels or SR policy, exist between the App-aware-process Head-End and the App-aware-process End-Point. The App-aware-process Head-End maintains the matching relationship between the application characteristic information and the paths between the App-aware-process Head-End and the App-aware-process End-Point. The App-aware-process Head-End determines the path between the App-aware-process Head-End and the App-aware-process End-Point according to the application characteristic information carried in the packets and the matching relationship with it, which satisfies the service requirements of the application. If there is no such matching path found, the App-aware-process Head-End can set up a path towards the App-aware-process End-Point, and the matching relationship will be stored. The App-aware-process Head-End forwards the packets along the path. The application information conveyed by the packet received from the App-aware Edge Device can also be copied or be mapped to the out IPv6 extension header for further application-aware process.

4. App-aware-process Mid-Point: The Mid-Point provides the path service according to the path set up by the App-aware-process Head-End which satisfies the service requirements conveyed by the IPv6 packets. The Mid-Point may also adjust the resource locally to guarantee the service requirements depending on a specific policy and the application-aware information conveyed by the packet. Policy definitions and mechanisms are out of the scope of this document.

5. App-aware-process End-Point: The process of the specific service path will end at the End-Point. The service requirements information can be removed at the End-Point together with the outer IPv6 encapsulation or go on to be conveyed with the IPv6 packets.

In this way the network is aware of the service requirements expressed by the applications explicitly. According to the service requirement information carried in the IPv6 packets the network is able to adjust its resources fast in order to satisfy the service

requirement of applications. The flow-driven method also reduces the challenges of inter-operability and long control loop.

5. APN6 Requirements

Utilizing IPv6 encapsulation (e.g. IPv6 header as well as, possibly, extension headers), the application-aware information is conveyed into the network which performs service provisioning, traffic steering, and SLA guarantee according to such information. This section specifies the requirements for supporting the APN6 framework, including the requirements for conveying and handling the application-aware information and related security requirements.

5.1. Application-aware Information Conveying Requirements

The application-aware information includes application-aware identification information and network performance requirements information.

1. Application-aware identification information includes the following identifiers (IDs),

   * SLA level: indicating the level of SLA requirement of the application such as Gold, Silver, Bronze. In some cases, color (e.g. red, green) can be used to indicate the SLA level.

   * Application ID: identifying an application.

   * User ID: identifying the user of the application.

   * Flow ID: identifying the flow which the application traffic belongs to.

   The different combinations of the IDs can be used to provide different granularity of the service provisioning and SLA guarantee for the traffic.

2. Network performance requirements information includes the following parameters,

   * Bandwidth: the bandwidth requirement of the application traffic

   * Latency: the latency requirement of the application

   * Jitter: the jitter requirement of the application

The different combinations of the parameters are for further expressing the more detailed service requirements of an application, conveyed together with the Application-aware identifiers, which can be used to match to appropriate tunnels/SR Policies, queues that can satisfy these service requirements. If not available, new tunnels/SR Policies can also be triggered to be set up.

[REQ 1a]. Application-aware identification information MUST include Application ID to indicate the application that generates the packet.

[REQ 1b]. SLA level is RECOMMENDED to be included in the Application-aware identification information.

[REQ 1c]. User ID and Flow ID are OPTIONAL to be included in the Application-aware identification information.

[REQ 1d]. Network performance requirements information is OPTIONAL.

[REQ 1e]. All the nodes along the path SHOULD be able to process the application-aware information if needed.

[REQ 1f]. The application-aware information can be generated directly by application, or by the application-aware edge devices though packet inspection or local policy.

[REQ 1g]. The application-aware information SHOULD be kept intact when directly copied from the application-aware edge devices and carried in the IPv6 encapsulation.

5.2. Application-aware Information Handling Requirements

The app-aware-process Head-End and app-aware-process Mid-Point perform matching operation against the application-aware information, that is, to match IDs and/or service requirements to the corresponding network resources (tunnels/SR policies, queues).

5.2.1. App-aware SLA Guarantee

In order to achieve better Quality of Experience (QoE) of end users and engage customers, the network needs to be able to provide fine-granularity and even application-level SLA guarantee [I-D.li-apn6-problem-statement-usecases].

[REQ 2-1a]. With the application-aware information, the App-aware-process Head-End SHOULD be able to steer the traffic to the tunnel/SR policy that satisfies the matching operation.

[REQ 2-1b]. With the application-aware information, the App-aware-process Head-End SHOULD be able to trigger the setup of the tunnel/SR policy that satisfies the matching operation.

[REQ 2-1c]. With the application-aware information, the App-aware-process Head-End and Mid-Point SHOULD be able to steer the traffic to the queue that satisfies the matching operation.

[REQ 2-1d]. With the application-aware information, the App-aware-process Head-End and Mid-Point SHOULD be able to trigger the configuration of the queue that satisfies the matching operation.

5.2.2. App-aware network slicing

Network slicing provides ways to partition the network infrastructure in either control plane or data plane into multiple network slices that are running in parallel. The resources on each node need to be associated to corresponding slices.

[REQ 2-2a]. With the application-aware information, the App-aware-process Head-End SHOULD be able to steer the traffic to the slice that satisfies the matching operation.

[REQ 2-2a]. With the application-aware information, the App-aware-process Mid-Point SHOULD be able to associate the traffic to the resources in the slice that satisfies the matching operation.

5.2.3. App-aware deterministic networking

Along the path each node needs to provide guaranteed bandwidth, bounded latency, and other properties relevant to the transport of time-sensitive data for the Detnet flows that coexist with the best-effort traffic.

[REQ 2-3a]. With the application-aware information, the App-aware-process Head-End SHOULD be able to steer the traffic to the appropriate path that satisfies the matching operation.

[REQ 2-3b]. With the application-aware information, the App-aware-process Head-End SHOULD be able to trigger the setup of the appropriate path that satisfies the matching operation for the Detnet flows.

[REQ 2-3c]. With the application-aware information, the App-aware-process Mid-Point SHOULD be able to associate the traffic to the resources along the path that satisfies the performance guarantee.

[REQ 2-3d]. With the application-aware information, the App-aware-process Mid-Point SHOULD be able to reserve the resources for the Detnet flows along the path that satisfies the performance guarantee.

5.2.4. App-aware service function chaining

The end-to-end service delivery often needs to go through various service functions, including traditional network service functions such as firewalls, DPI as well as new application-specific functions, both physical and virtual. SFC is applicable to both fixed and mobile networks as well as data center networks.

[REQ 2-4a]. With the application-aware information, the App-aware-process devices SHOULD be able to steer the traffic to the appropriate service function.

[REQ 2-4b]. The App-aware-process devices SHOULD be able to process the application-aware information carried in the packets.

5.2.5. App-aware network measurement

Network measurement can be used for locating silent failure and predicting QoE satisfaction, which enables real-time SLA awareness/ proactive OAM.

[REQ 2-5a]. With the application-aware identification information, the App-aware-process devices SHOULD be able to perform IOAM based on the Application ID.

[REQ 2-5a]. With the application-aware information, the network measurement results can be reported based on the Application ID and verify whether the performance requirements of the application are satisfied.

5.3. Security requirements

[REQ 3a]. The security mechanism defined for APN6 MUST allow an operator to prevent applications sending arbitrary application-aware information without agreement with the operator.

[REQ 3b]. The security mechanism defined for APN6 MUST prevent an application requesting a service that is not entitled to get.

6. IANA Considerations

This document does not include an IANA request.

7. Security Considerations

[I-D.li-apn6-problem-statement-usecases] and section 5.3 describe the security considerations and requirements for APN6.

8. Acknowledgements

The authors would like to acknowledge Robert Raszuk (Bloomberg LP) and Yukito Ueno (NTT Communications Corporation) for their valuable reviews and comments.

9. Contributors

Liang Geng
China Mobile
China

Email: gengliang@chinamobile.com

Chang Cao
China Unicom
China

Email: caoc15@chinaunicom.cn

Cong Li
China Telecom
China

Email: licong.bri@chinatelecom.cn

10. References

10.1. Normative References

[I-D.li-apn6-problem-statement-usecases]
Li, Z., Peng, S., Voyer, D., Xie, C., Liu, P., Liu, C., Ebisawa, K., Ueno, Y., Previdi, S., and J. Guichard, "Problem statement and use cases of Application-aware IPv6 Networking (APN6)", draft-li-apn6-problem-statement-usecases-00 (work in progress), September 2019.

[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>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <https://www.rfc-editor.org/info/rfc8578>.

10.2. Informative References

[RFC3272] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, DOI 10.17487/RFC3272, May 2002, <https://www.rfc-editor.org/info/rfc3272>.

Authors' Addresses

Zhenbin Li
Huawei Technologies
China

Email: lizhenbin@huawei.com

Shuping Peng
Huawei Technologies
China

Email: pengshuping@huawei.com

Daniel Voyer
Bell Canada
Canada

Email: daniel.voyer@bell.ca

Chongfeng Xie
China Telecom
China

Email: xiechf.bri@chinatelecom.cn

Peng Liu
China Mobile
China

Email: liupengyjy@chinamobile.com

Chang Liu
China Unicom
China

Email: liuc131@chinaunicom.cn

Kentaro Ebisawa
Toyota Motor Corporation
Japan

Email: ebisawa@toyota-tokyo.tech

Stefano Previdi
Individual
Italy

Email: stefano@previdi.net

James N Guichard
Futurewei Technologies Ltd.
USA

Email: jguichar@futurewei.com