

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 3, 2022

Z. Du
P. Liu
China Mobile
March 2, 2022

Auto-adjustment of Encapsulation Information in APN6
draft-du-apn6-auto-encapsulation-adjustment-02

Abstract

This document introduces a method to adjust the encapsulation information in Application-aware IPv6 Networking.

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 September 3, 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

Internet-Draft

Auto-adjusted APN6 Encapsulation

March 2022

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	2
2.	Auto-adjustment of Encapsulation Information	2
2.1.	Current Mechanism in APN6	3
2.2.	Comparisons of Data Plane and Control Plane Programming .	3
2.3.	Potential Solutions for Auto-adjustment	4
3.	IANA Considerations	5
4.	Security Considerations	5
5.	Acknowledgements	5
6.	References	5
6.1.	Normative References	5
6.2.	Informative References	5
	Authors' Addresses	6

[1.](#) Introduction

As the development of 5G and the new emerging Internet services, such as live video streaming, the networks are facing a larger and larger SLA requirement difference. For better bearing of the user's traffic, the networks need to be intelligent and be aware of the user traffic's demand. An innovative method called APN6 is introduced in [[I-D.li-apn6-problem-statement-usecases](#)] and [[I-D.li-apn-framework](#)].

In the mechanism of APN6, the packet can carry the ID information and SLA requirements of the traffic, and a network equipment can get them in each packet and handle the packet accordingly. It is one kind of network programming mechanisms in the data plane.

As the encapsulation information increases in an APN packet, some bandwidth is kindly wasted in APN6 which contains a larger overhead in every packet. On one aspect, it is believed that it is necessary for the evolution to an intelligent network; on the other aspect, it is recommended that after the network has known the requirements of the traffic and associated it with a proper policy, the traffic does not need to resend the same information in every packet again and again. This document describes the process of the later.

[2.](#) Auto-adjustment of Encapsulation Information

2.1. Current Mechanism in APN6

As shown in Figure 1, the APN framework [[I-D.li-apn-framework](#)] includes Service-aware App, App-aware Edge Device, App-aware-process Head-End, App-aware-process Mid-Point, and App-aware-process End-Point.

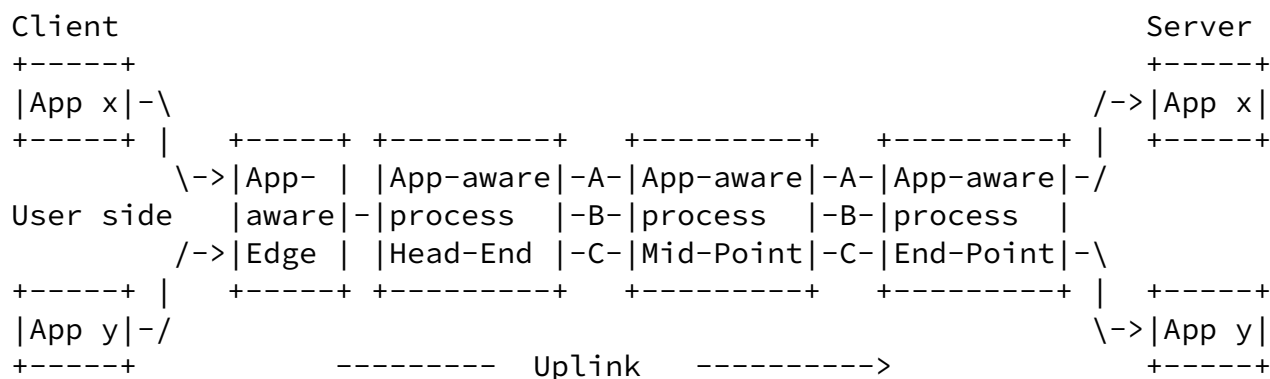


Figure 1: Framework and Key Components in APN6

The data-driven process of APN6 is described below.

The APP or the APP-aware Edge will generate APN packets each carries the application characteristic information in the encapsulation.

App-aware-process Head-End can read that information and steer the packets into a given policy which satisfies the application's SLA requirements. It is supposed that a set of paths, tunnels or SR policies, exist between the App-aware-process Head-End and the App-aware-process End-Point. App-aware-process Head-End can find one existing path or establish a new one for the traffic.

2.2. Comparisons of Data Plane and Control Plane Programming

We can realize the same traffic steering in the control plane. The

control-plane based process, as described below, includes three key components: the identity of the traffic, policies in Head-End, and the interface to notify the user requirements.

The APP or the Edge knowing the application characteristic information, needs to report that information to the controller of the Head-End by some means.

The controller needs to know the traffic requirements and the status of the network, and generate a policy for the Head-End. The policy SHOULD include the identity of the traffic and the path that the traffic should follow.

The Head-End needs to implement the policy, and steer the traffic to the proper path.

In this mechanism, we do not need to carry extra information in each packet, but need to generate control messages between the Edge and the controller, and between the Head-End and the controller.

In the situation that the traffic is small, and simple to handle, a control-layer decision-loop is not that necessary. By comparison, a date-driven method is more flexible. In this situation, the Head-End after steering the traffic needs to report the (summarized) change to the controller.

[2.3.](#) Potential Solutions for Auto-adjustment

We can find that after the Head-End has selected the policy, the extra information carried in the following APN6 packets has little use. Therefore, an auto-adjustment of encapsulation information mechanism may be helpful for the simplification of the following IPv6 packets.

According to [[I-D.li-apn-framework](#)], the information may include application-aware identification, such as SLA level, application ID, user ID, flow ID, etc., and network performance requirements, such as bandwidth, latency, jitter, packet loss ratio, etc. Hence, at least, we can send only the application-aware identification information in the following APN6 packets without network performance requirements information.

Two methods are described below.

One straightforward method is that we firstly send full information in APN6 packets, and after several seconds, we send APN6 packets that only contain the necessary information, such as the application-aware identification information. In this method, we believe that the Head-End can handle the policy mapping process in the several seconds. For example, it can be three seconds. The number should be a parameter that can be adjusted according to the situation of each network.

Another method is that after enabling the policy, the Head-End can notify the encapsulation point by some means. However, we do not have a notification mechanism between different nodes in the data-plane network programming now. We need to notify by using the control plane again. The control plane sends a message to the encapsulation point to adjust the encapsulation degree.

This document suggests to enable a simple notification method for the data-plane network programming if the information is not that complicated. For example, we can send a "ping" message with a specific flag to the encapsulation point. The advantage is easy to inter-operate.

In future, with the technical development of network equipments, the bandwidth may not be the bottleneck anymore, so that a full APN6 encapsulation packet may be used widely to enable the data plane intelligence. However, the auto-adjustment of encapsulation information method can help the adoption of the APN6 mechanism by providing a transit solution. Meanwhile, this document also provides a feedback mechanism for the data plane programming to enable the coordination between two nodes.

[3.](#) IANA Considerations

TBD.

[4.](#) Security Considerations

TBD.

[5.](#) Acknowledgements

TBD.

[6.](#) References

[6.1.](#) Normative References

[I-D.li-apn-framework]

Li, Z., Peng, S., Voyer, D., Li, C., Liu, P., Cao, C., Mishra, G., Ebisawa, K., Previdi, S., and J. N. Guichard, "Application-aware Networking (APN) Framework", [draft-li-apn-framework-04](#) (work in progress), October 2021.

[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>>.

[6.2.](#) Informative References

[I-D.li-apn6-problem-statement-usecases]

Li, Z., Peng, S., Voyer, D., Xie, C., Liu, P., Liu, C., Ebisawa, K., Previdi, S., and J. N. Guichard, "Problem Statement and Use Cases of Application-aware IPv6 Networking (APN6)", [draft-li-apn6-problem-statement-usecases-01](#) (work in progress), November 2019.

Authors' Addresses

Zongpeng Du
China Mobile
No.32 XuanWuMen West Street
Beijing 100053
China

Email: duzongpeng@foxmail.com

Peng Liu
China Mobile
No.32 XuanWuMen West Street
Beijing 100053
China

Email: liupengyjy@chinamobile.com