Workgroup: Network Working Group
Internet-Draft:
draft-jiang-service-oriented-ip-023
Published: 15 May 2020
Intended Status: Informational
Expires: 16 November 2020

Authors: B. E. Carpenter   S. Jiang
Univ. of Auckland   Huawei Technologies Co., Ltd
G. Li
Huawei Technologies

Service Oriented Internet Protocol

Abstract

This document proposes a new, backwards-compatible, approach to packet forwarding, where the service required rather than the IP address required acts as the vector for routing packets at the edge of the network. Deeper in the network, the mechanism can interface to conventional and future methods of service or application aware networking.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

An important aspect of the Internet today is that it is no longer a uniform space with uniform requirements. For both technical and economic reasons, we see an emerging trend of usage scenarios that are confined to some form of limited domain, and which inevitably lead to applications and protocols that are only suitable within a given scope [I-D.carpenter-limited-domains]. In particular, various techniques have emerged for packet treatments that are specific to a type of application or service. This trend collides immediately with two factors: the original design concept of an Internet with end to end IP transparency (such that any locally defined protocol running over IP is almost certain to escape the local network), and with the increasing presence of middleboxes. In this emerging context, where end to end IP service is no longer a safe assumption, and where there is increasing demand for specific services, this document proposes a new, backwards-compatible, approach to packet forwarding, where the service required rather than the IP address required acts as the vector for routing packets at the edge of the network, close to the host requiring a particular service or application. This form of service based packet forwarding is referred to as Service Oriented IP (SOIP).

We propose an addition to the existing core function of the network, which is reachability over IPv6 or IPv4. Today, IP is focussed on reachability, using best effort forwarding both to find a route and to automatically share transmission resources in a simple and low-cost way. As a result, transport protocols such as TCP and UDP learn little or nothing about the network, beyond congestion or loss signals. Several ISPs may lie on the path between a user and a server, but they are largely ignorant about the services a user requires. Typical services could be streamed content, regular content, user posting, storage access, or calculation, but this list is not exclusive.

Both service providers and users will benefit if a packet stream can be identified intrinsically as requiring a certain kind of service. This is particularly applicable for edge networks, such as those supported by 5G technology, where there is an emphasis on upper layer service provision. Whatever the business model - for example the ISP operates all types of service, or the ISP operates no user services at all and has contracts with specific service providers, or the ISP is agnostic about user services - SOIP will allow for optimised packet delivery. The ISP will have the choice to provide some or all services. The user will have the choice to use ISP services or bypass them. Traffic that leaves the domain where SOIP is in use will be perfectly normal IPv6 or IPv4 traffic, sent by an exit node acting as a proxy (not an IP-layer translator) for the user. Additionally, IPv6 and IPv4 will be modelled as services available to the user, thereby giving continuity of access to everything the user has today. This is a logical extension of a principle already adopted to model IPv4 as a service available via IPv6 [RFC8585].

As new service and application oriented features are deployed in the network, SOIP will provide a seamless interface to both existing and future mechanisms. Effectively it will make client hosts future-proof as the network evolves.

2. Proposed Solution

NOTE: This is a preliminary draft expected to stimulate discussion, so many details are not yet defined.

The approach is to make the service be the central component of the network, from the end user's point of view. Conceptually, the user's packets will be directed at a service, not at an IP host. The first hop SOIP router will either forward the packets to an upstream SOIP router, or immediately dispatch the session to a suitable service. At least one SOIP router in a domain must be capable of acting as a dispatcher. The dispatched traffic may either remain in SOIP format, or be transformed by a proxy mechanism into a conventional IP-based format. Figure 1 gives an overview, and Section 5 and Section 6 discuss this further.

```
-----------         -----------
|End-user | <-----> |  SOIP   |
|  Host   |         | router  |
-----------         -----------
                         ^
                         |
                         v
   --------------   -----------    -----------------
   | SOIP-based |   | SOIP      |   | Unknown         |
   | services   |---| router + |---| future          |
   |            |   |dispatcher|   | services        |
   --------------   ------------   -----------------
                     |    |   |
   --------------    |    |   |    -----------------
   |Traditional |____/    |   \____|Segment Routing|
   |IP services |         |        |services       |
   --------------   ------------   -----------------
                    |   SDN    |
                    | services |
                    ------------
```

Figure 1

The service actions that the network can provide are abstracted into a number of classes called Service Action Types (SATs). While there needs to be flexibility and extensibility, the number of service action types will be limited. They will not be numerous like IP protocol numbers or well-known TCP or UDP port numbers. Along with the SAT, a source IPv6 address is used to identify the client system. As will be seen below, the destination IPv6 address becomes optional. A consequence is that the IP header and some aspects of the protocol stack have to be redesigned. We will show below how this can be done without disturbing most of the running network. Another consequence is that the first step in processing a packet is to process the SAT, not the destination address (if there is one).

Traditional reachability, when required, is provided by classical IPv6, or by IPv4 as-a-service.

When an SOIP packet enters a router, it is classified at line speed according to the SAT. Routing to upstream SOIP routers will be based on the SAT, not on a destination IP address. Routing from a dispatcher may be based on the SAT if the service required is based on SOIP, or on a conventional IP address otherwise. A preliminary list of SATs is shown in Figure 2, with brief descriptions:

SATs	Direction	Description
IPv4 reachability	Request	IPv4 destination host
	Response	IPv4 source host
IPv6 reachability	Request	IPv6 destination host
	Response	IPv6 source host
Discovery Service	Request	Discover network services, e.g. DNS, CDN. May map to IP Anycast
	Response	Content ID, service ID. Or "service not available" error
Multicast Service	Request	Join multicast service for some content, e.g. video stream
	Response	Multicast directory answers request, provides m/c source.
Computation Service	Request	Submit task to network, with computation type, task descriptor and requester ID
	Response	Computation resource ID, or direct result of task. Or "service not available" error
Storage Service	Request	Submit/retrieve data to/from network storage, with data description and/or data ID
	Response	Storage locator, data ID. Or "service not available" error
App Server Service	Request	To submit source code or deploy package to application platform, with necessary configurations.
	Response	Answer with service ID. Or "service not available" error

Figure 2

For each request there will be a corresponding response. The details remain to be worked out - probably a generic response message including the SAT. To allow multiple overlapping sessions, each request/response sequence should have a unique ID, which will be used by the SOIP dispatcher to match service responses to the appropriate user session.

For IPv6-only networks, is expected that IPv4 reachability will be provided by a solution such as 464XLAT. Also, no separate SAT is needed for IPv6 to IPv4 translation. For example, if a host requests IPv4 reachability but supplies an IPv6 address as its own locator, NAT64 [RFC6146] is implied.

For the moment codes for the SATs are undefined, but they are assumed to be small integers. There are two possible approaches to the packet format. One is to use a traditional Type-Length-Value (TLV) layout. Another is to use a more flexible encoding at the lowest level, taking advantage of some form of network processor in the routers. An obvious choice would be Concise Binary Object Representation (CBOR) [RFC7049], which combines flexibility with efficiency. In either case we could require that the first four bits of the wire format are a new IP version number other than 4 or 6. An alternative, at least for early experimentation, is to run SOIP over UDP and IPv6.

Examples of both encoding choices are described below. In either case, the essential content of a packet header is as follows:

- The SAT code (small integer)

- Flag bits

- Traffic class (as for IPv6)

- Session Identifier (so that sessions can be tracked regardless of IP address)

- Hop limit (small integer)

- User locator (IP address or identifier)

- Service data length (not needed in CBOR version)

- Service data (length depends on SAT)

- Payload length (not needed in CBOR version)

- Payload

Experience with IPv4 options, and IPv6 extension headers and options, has shown that new ones are very hard to deploy on an operational network, and that the ones defined during the initial design are not always useful. Therefore we propose that all options and extensions are defined as part of the service data and are not visible as part of the basic packet header, giving good flexibility.

We propose to include the exact equivalent of the IPv6 Traffic Class [RFC8200], which can work exactly as for IPv6. In contrast, one defect in the IPv6 flow label [RFC6437] is that it is different in the two directions of a flow. Instead we propose a session ID that is the same in both directions, which has various advantages by allowing immediate session identification.

The flag bits provide useful indications to the routing system, if set:

- *Mobile - set if the user system is mobile

- *Flow size (3 bits)

  - -000 means a single packet, no flow/congestion state needed

  - -other values TBD indicate type of flow/congestion state

- *Authenticated - set if packet authenticated (details TBD)

- *Encrypted - set if encryption applied (details TBD)

Note that fragmentation is not supported. Fragmentation, and the related mechanisms of MTU discovery, are a significant operational problem in the current Internet [I-D.ietf-intarea-frag-fragile]. We simply abolish this problem area in SOIP, which is designed for use in managed networks where a single size of maximum transmission unit (MTU) is available everywhere. An SOIP network will have a globally defined MTU. Of course, IPv4 and IPv6 reachability services via the open Internet will have to support PMTUD and fragmentation as best they can, but this concerns the embedded IP packets, not the SOIP packets, and will be invisible locally.

Appendix A outlines possible TLV and CBOR encodings of the SOIP protocol.

3. Coexistence Issues

SOIP is expected to coexist with IPv6; in a sense it is a low-level method of orchestrating IPv6 connections. We assume that each SOIP client host has at least one IPv6 address, and that SOIP routers will announce themselves using a suitable IPv6 Router Advertisement extension [I-D.troan-6man-universal-ra-option]. Normally, the first-hop SOIP router will be the same as the IPv6 first-hop router.

As a result of this, all standard management mechanisms such as NETCONF may be used without further specification. Also, when a data connection of any kind is established after a SOIP request/response exchange, all standard transport mechanisms are available over IPv6. As noted above, they are subject to the locally defined MTU as long as they remain within the SOIP domain.

We do not define how SOIP would operate in an IPv4-only network.

4. Some Usage Examples

- *Storage request (upload content):

  - -Service data identifies storage requirement (temporary/permanent, private/public, encryption, etc.)

  - -Payload identifies data (path/name.format, etc.)

- *Storage request (download content):

  - -Service data identifies transmission requirement (streamed, block, etc. and the specific transport protocol - UDP, TCP, QUIC, etc. - if needed)

  - -Payload identifies data (path/name.format, etc.)

- *Computation request

  - -Service data identifies computing requirement

  - -Payload identifies computing application

- *Reachability request

  - -Service data gives destination IP address (or DNS name)

  - -Indication of transport protocol required (details TBD)

  - -Indication of options or extension headers required (details TBD)

5. Continuity with the Existing Internet

Continuity is provided in two ways:

1. A user node can simply use IP completely in parallel with SOIP. The network stack in the user node will simply encode the IP

packets as SOIP packets with the SAT for IP reachability, and a SOIP dispatcher will send and receive IP packets at the SOIP domain boundary.

2. If a service in the SOIP domain needs service from elsewhere in the IP Internet to respond to a user request, it will use a similar dispatcher function to do so.This could also be described as a proxy mechanism. (Of course, services in interconnected SOIP domains may talk to each other directly.)

6. Interface with Service and Application Domains

Various techniques are emerging for service or application specific networking within operators' networks. An overview of the motivations is given in [I-D.li-apn6-problem-statement-usecases], and specific techniques have been defined such as Network Service Headers [RFC8300] and Segment Routing [RFC8402], as well as Software-Defined Networking in general [RFC7426]. A SOIP dispatcher that is aware of such techniques may convert SOIP traffic into one of these mechanisms, for example by encapsulation or proxying. Furthermore, the model is future-proof. The dispatcher could be upgraded to support unknown future service or application oriented networking mechanisms, without requiring changes to SOIP clients or routers.

7. Security Considerations

It is intended that both authentication and encryption should be available for all SOIP packets. However, this requires further work, especially to determine whether existing mechanisms for key management can be used.

Since clients are identified by an IPv6 address, existing layer 3 privacy considerations for IPv6 addresses will apply to SOIP [RFC7721]. Upper layer privacy considerations will depend on the service concerned.

8. IANA Considerations

This document makes no request of the IANA.

9. References

[I-D.carpenter-limited-domains]
Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", Work in Progress, Internet-Draft, draft-carpenter-limited-domains-13, 2 February 2020, <https://tools.ietf.org/html/draft-carpenter-limited-domains-13>.

[I-D.ietf-intarea-frag-fragile]
Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", Work in Progress, Internet-Draft, draft-ietf-intarea-frag-fragile-17, 30 September 2019, <https://tools.ietf.org/html/draft-ietf-intarea-frag-fragile-17>.

[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. Guichard, "Problem Statement and Use Cases of Application-aware IPv6 Networking (APN6)", Work in Progress, Internet-Draft, draft-li-apn6-problem-statement-usecases-01, 3 November 2019, <https://tools.ietf.org/html/draft-li-apn6-problem-statement-usecases-01>.

[I-D.troan-6man-universal-ra-option]
Troan, O., "The Universal IPv6 Configuration Option (experiment)", Work in Progress, Internet-Draft, draft-troan-6man-universal-ra-option-02, 2 April 2020, <https://tools.ietf.org/html/draft-troan-6man-universal-ra-option-02>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <https://www.rfc-editor.org/info/rfc6146>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <https://www.rfc-editor.org/info/rfc7049>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <https://www.rfc-editor.org/info/rfc7426>.

[RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <https://www.rfc-editor.org/info/rfc7721>.

[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>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8585] Palet Martinez, J., Liu, H. M.-H., and M. Kawashima, "Requirements for IPv6 Customer Edge Routers to Support IPv4-as-a-Service", RFC 8585, DOI 10.17487/RFC8585, May 2019, <https://www.rfc-editor.org/info/rfc8585>.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/info/rfc8610>.

Appendix A. Possible TLV and CBOR Encodings

A.1. TLV Mapping

Figure 3 shows a possible type-length-value packet format.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|R R R R|      SAT      |M F F F A E R R| Traffic Class |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Session Identifier  (32 bits)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Hop Limit   |                                               |
+-+-+-+-+-+-+-+-+                                               +
|                                                               |
+                                                               +
|              Client Locator / Identifier  (128 bits)          |
+                                                               +
|                                                               |
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               | SD length     |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
|                                                               |
+              Service Data (variable length)                   +
...                                                           ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Payload Length         |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|                               +
|                                                               |
+              Payload Data (variable length)                   +
...                                                           ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3

Version - IP version number TBD (not needed over UDP)

R - Reserved, must be zero (not needed over UDP)

SAT - Service Action Type code

M - Mobile flag

FFF - Flow type flags (FFF = 000 for single-packet flows; other values for longer flows)

A - Authentication flag

E - Privacy (encryption) flag

Traffic class, exactly as for IPv6

Session identifier, 32-bit pseudo random number

Client Locator / Identifier - globally unique IPv6 address

A.2. CBOR Mapping

The packet consists of a CBOR byte string preceded by a single byte (Figure 4). For example, for version 7, this byte would be 0x70. This byte is not decoded as CBOR, and is not needed over UDP.

```
+-+-+-+-+-+-+-+-+
|Version|R R R R|
+-+-+-+-+-+-+-+-+
```

Figure 4

The CBOR bytes then obey the CDDL [RFC8610] specification in Figure 5.

```
sat-packet = [sat, flags, traffic-class, session-id, hop-limit,
              source, service-data, ?payload]

sat = 0..255
flags = bytes .size 1
traffic-class = 0..255
session-id = 0..4294967295 ;up to 32 bits
hop-limit = 0..255
client = ipv6-address
service-data = any
payload = any

ipv6-address = bytes .size 16
```

Figure 5

The syntax of the various service-data formats can be defined in separate documents for each SAT value.

We assume that routers capable of handling a CBOR-based layer 3 protocol will exist, and will use some form of programmable network processor rather than traditional ASIC or FPGA designs. This allows great flexibility and software-friendly extensibility, especially of the service data formats. Further investigation is needed whether this is realistic.

Appendix B. Change log [RFC Editor: Please remove]

- *draft-jiang-service-oriented-ip-00, 2019-05-07:

  - -Initial version

- *draft-jiang-service-oriented-ip-01, 2019-06-21:

  - -Editorial corrections

- *draft-jiang-service-oriented-ip-02, 2019-10-29:

  - -Added overview diagram

  - -Added discussion of dispatcher function

  - -Clarifications and editorial corrections

- *draft-jiang-service-oriented-ip-03, 2020-05-15:

  - -Editorial corrections

  - -Converted to xml2rfc v3

Authors' Addresses

Brian Carpenter
The University of Auckland
School of Computer Science
University of Auckland
PB 92019
Auckland 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing, 100095
P.R. China

Email: jiangsheng@huawei.com

Guangpeng Li
Huawei Technologies
Q14, Huawei Campus
No.156 Beiqing Road
Hai-Dian District, Beijing
100095
P.R. China

Email: liguangpeng@huawei.com