

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 3 January 2022

J. Arkko  
J. Novotny  
Ericsson  
2 July 2021

Privacy Improvements for DNS Resolution with Confidential Computing  
draft-arkko-dns-confidential-02

Abstract

Data leaks are a serious privacy problem for Internet users. Data in flight and at rest can be protected with traditional communications security and data encryption. Protecting data in use is more difficult. In addition, failure to protect data in use can lead to disclosing session or encryption keys needed for protecting data in flight or at rest.

This document discusses the use of Confidential Computing, to reduce the risk of leaks from data in use. Our example use case is in the context of DNS resolution services. The document looks at the operational implications of running services in a way that even the owner of the service or compute platform cannot access user-specific information produced by the resolution process.

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 3 January 2022.

Copyright Notice

Copyright (c) 2021 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 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. Background . . . . .	4
3. Terminology . . . . .	5
4. Prerequisites . . . . .	5
5. Confidential Computing . . . . .	6
6. Using Confidential Computing for DNS Resolution . . . . .	7
7. Operational Considerations . . . . .	9
7.1. Operations . . . . .	9
7.2. Debugging . . . . .	11
7.3. Dependencies . . . . .	11
7.4. Additional services . . . . .	12
7.5. Performance . . . . .	12
8. Security Considerations . . . . .	13
8.1. Observations from outside the TEE . . . . .	13
8.2. Trust Relationships . . . . .	13
8.3. Denial-of-Service Attacks . . . . .	14
8.4. Other vulnerabilities . . . . .	15
9. Recommendations . . . . .	16
10. Acknowledgments . . . . .	17
11. References . . . . .	17
11.1. Normative References . . . . .	17
11.2. Informative References . . . . .	17
Authors' Addresses . . . . .	22

## 1. Introduction

DNS privacy has been a popular topic in the last few years, and continues to be. The issues with regards to privacy are first that domain name meta-data is visible on the wire, even when the actual communications are encrypted. This is being addressed with better technology.

But even if the meta-data is hidden inside communications, any DNS resolvers still have the potential too see users' entire browsing history. This is particularly problematic, given that commonly used large public or operator resolver services are an obviously

attractive target, for both attacks and for commercial or other use of information visible to them.

A lot of work is ongoing in the industry and the IETF to address some of these issues:

- \* Work on encrypted DNS query protocols to hide the meta-data related to domain names.
- \* Discovery mechanisms. These may enable a bigger fraction of DNS query traffic to move to encrypted protocols, and may also help distributed queries to different parties to avoid concentrating all information in one place.
- \* Practices, expectations, contracts (e.g., [RFC8932], Mozilla's trusted recursive resolver requirements [MozTRR])
- \* Improvements outside DNS (e.g., encrypted Server Name Indication (eSNI) [I-D.ietf-tls-esni]).
- \* General technology developments (e.g., confidential computing, attestations, remote attestation work at the IETF RATS WG, and so on)

The goal of this document is to build on all that work - and assume all communications are or become encrypted, including the DNS traffic. Our question is what problems remain? Is there a next step?

Our worry is that resolvers can be a major remaining source of leaks, e.g., through accidents, attacks, commercial use, or requests from the authorities. We need to protect user's data in flight, at rest, or in use - we wanted to experiment with technology that could reduce leaks on the last two cases. Confidential Computing is one such potential technology, but it is important to talk about it and get broader feedback. The use of this technology does have some operational impacts.

Our primary conclusions are that data held by servers should receive at least as much security attention as communications do. The authors feel that this is particularly crucial for DNS, due to the potential to leak of users' browsing histories, but principles apply also to other services.

As a result, all applicable tools should be considered, including confidential computing that is discussed in this document. However, the operational and business implications of such tools should be considered. Feedback to us is very welcome. Are these approaches

feasible or infeasible? What aspects need to be taken into account to successfully apply them?

## 2. Background

Communications security has been at the center of many security improvements in the Internet. The goal has been to ensure that communications are protected against outside observers and attackers [RFC3552] [RFC7258]. Communications security is, however, not sufficient by itself, and continuing success in better protection of communications is highlighting the need to address other issues.

In particular, more attention needs to be paid to protecting data not just in flight but also at rest or in use. User data leaks that can occur from servers and other systems, through accidents, attacks, commercial use of data, and requests for information by authorities. Both data at rest and data in use needs to be protected. Being able to protect data in use provides also benefits to protecting keys used for protecting data in flight and at rest.

Data leaks are very common, and include highly publicized ones or ones with significant consequences, such as [Cambridge]. Data leaks are also not limited to traditional computer applications, but can also impact anything from private health data [Vastaamo] to children's toys [Toys] or smart TVs [SmartTV].

The general issue and possible solutions have been discussed extensively elsewhere, e.g., [Digging], [Mem], [Comparison], [Innovative], [AMD], [Efficient], [CCC-Deepdive], [CC], and so on. The Internet-relevant angle has also been discussed in few documents, e.g., [I-D.lazanski-smart-users-internet], [I-D.iab-dedr-report] [I-D.arkko-farrell-arch-model-t-redux], and so on. The topic is also related to best practices for protocol and network architecture design, and what information can be provided to what participants in a system, see, e.g. [RFC8558] [I-D.thomson-tmi] [I-D.arkko-arch-infrastructure-centralisation].

Data leaks can occur in user-visible services that user has chosen to use and agreed to provide information to (at least in theory [Unread]). But leaks can also occur in other types of services, that are part of the infrastructure, such as DNS resolution services or parts of the communication infrastructure.

This document looks at the possibility of using a specific technical solution, Confidential Computing [CCC-Deepdive], to reduce the risk of leaks from data in use. We consider the operational implications of running services in a way that even the owner of the service or

compute platform cannot access user-specific information that is produced as a side-effect of the service.

We explore the use of Confidential Computing in the context of DNS resolution services [RFC1035]. This is a nice and relatively simple example, but there are of course potential other applications as well.

DNS resolution services are of course also an important case where privacy matters a lot for the users. Threats against the resolution process could prevent the user from accessing services. Data leaks from the process have the potential to expose the user's entire browsing history.

The use of Confidential Computing in the DNS context has been also discussed in other documents, e.g., [PDoT] and [I-D.reddy-add-server-policy-selection].

The DNS privacy issues have been also discussed in multiple documents, such as [RFC7626] [RFC8324] and so on.

### 3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

### 4. Prerequisites

The primary sources of leaks are as follows:

- \* Communications interception. This threat can be addressed by encrypted communications, such as the use of DNS-over-TLS (DoT) [RFC7858], DNS-over-HTTPS (DoH) [RFC8484], or DNS-over-QUIC (DoQ) [I-D.ietf-dprive-dnsquic] instead of traditional DNS protocols.
- \* Data leakage from the server or service, either from data at rest or in use. This can be addressed by encrypting the data while at rest and employing the techniques discussed in this document for data in use.

The specific information that is privacy sensitive depends on the application. In DNS resolution application it is clear that the users' browsing histories, i.e., which users asked for what names is privacy sensitive, and protecting that information is the primary focus in this document. In contrast, the domains themselves or the

associated address information is in the general case public and not privacy sensitive. However, in some cases even this information may be sensitive, such as in the case of internal domains of a corporate network. Information not related to individuals may also be sensitive in some cases, e.g., the collective browsing destinations of an entire organization.

The above was also observed in [RFC7626] which stated the following:

"DNS data and the results of a DNS query are public [...], and may not have any confidentiality requirements. However, the same is not true of a single transaction or a sequence of transactions; that transaction is not / should not be public."

Nevertheless, it should be noted that technology can help only insofar as there is commercial willingness to provide the best possible service and to protect the users' information.

Similarly, the techniques discussed in this document are not the sole, or full answer to all problems. There are a lot of technical, operational, and governance issues that also matter and practices that help. A good compilation of some best practices can be found in [RFC8932], and particularly Section 5.2 that discusses data at rest.

## 5. Confidential Computing

Confidential Computing is about protecting data in use by performing computation in a hardware enforced Trusted Execution Environment (TEE) [CCC-Deepdive]. It addresses the need to protect data in use, which traditionally has been hard to achieve. It may also help improve the encryption of data in flight and at rest, by helping protect session keys and other security information used in that process.

For our purposes, we focus on Trusted Execution Environments that use computer hardware to provide the following characteristics:

- \* **Attestability:** The environment can provide verifiable evidence to others (such as client using services running on it) about the environment, its characteristics, and the software it runs.
- \* **Code integrity:** Unauthorized entities cannot modify software being run within the environment.
- \* **Data confidentiality and integrity:** Unauthorized entities cannot view or modify data while it is in use within the TEE.

These characteristics have been paraphrased from [CCC-Deepdive]. See also [I-D.ietf-rats-architecture] for details of attestation. There are additional characteristics that matter in some situations, but for our purposes the above ones are central.

Specific technologies to perform Confidential Computing or run TEEs are becoming common in CPUs, operating systems, and other supporting software. For instance, Intel's Software Guard Extension (SGX) [SGX] is one CPU manufacturer's approach to this technology. SGX allows application developers to run software protected in a secure enclave protected by the CPU, including for instance encrypting all memory accesses outside the CPU and being able to provide remote attestation to outsiders about which software image is being run. These secure enclaves are the SGX approach to providing a TEE.

Confidential Computing is also becoming available on commonly available cloud computing services. When a user employs these services, they have the ability to run software and process data that even the owner of the cloud system does not have access to.

Interestingly, that is quite a contrast to the worries expressed some years ago about Trusted Computing technology, when it was feared that it enabled running software in users' computers that could act against the interests of the user in some cases, such as when protecting media files [Stallman]. While those concerns may apply even today in some cases, it is clear that whe the user can get secure information about services running somewhere in the network, this is an advantage for the users.

Note that availability might be another desirable characteristic for Confidential Computing systems, but it is one that is not in any special way supported by current technology. Ultimately, the owner of the computer still has the ability to choose when to switch the computer off, for instance. There is also no particular hardware technology at this time to deal with Denial-of-Service attacks. Some of the software techniques related to dealing with Denial-of-Service attacks are discussed in the Security Considerations section.

## 6. Using Confidential Computing for DNS Resolution

Confidential Computing can be used to provide a privacy-friendly resolution service in a server.

The basic arrangement is two-fold:

- \* User's computer and the DNS resolution server communicate using an encrypted and integrity protected transport protocol, such as DoT or DoH [RFC7858] [RFC8484].

- \* The secure connection terminates inside a TEE running in the the DNS resolution server. This TEE performs all the necessary processing to respond to the user's query. The TEE will not provide any user-specific information outside of the TEE, such as logs of what names specific clients queried for.

The TEE may need to contact other local servers or in the Internet to resolve a query that has no recently cached answer. We will discuss later how this can be done securely: it is necessary to prevent the linking any external actions such as receiving a client request and observing a query going out to other DNS servers in the Internet.

The arrangement is shown in Figure 1.

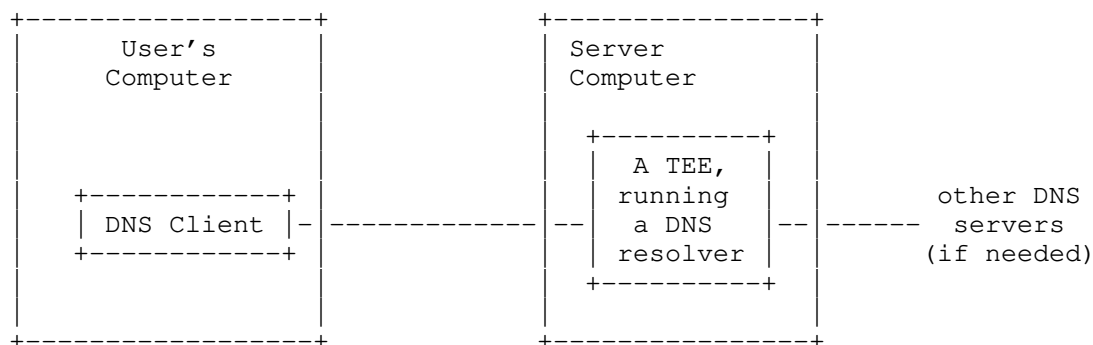


Figure 1: Confidential Computing for DNS Resolution

In this application, we strive to have no data at rest at all, at least nothing that relates directly to users. Data in flight and data in use are both protected by encryption. As a result of running the resolution service in this manner, any user-specific information should remain within the TEE, and not exposed to outsiders or even the owner of the service or the compute platform where the service is running in.

The authors believe that this is a desirable property. However, it remains to assure users and clients that the service is actually run in this manner. This can be done in two ways:

- \* Through off-line reliance on a particular service, i.e., a human decision to use a particular system. Once there is a decision to use a particular system, cryptographic means such as public keys may be used to ensure that the client is indeed connected to the expected server. However, there is no guarantee that the human-



space statements about the practices used in running the server are valid.

- \* Cryptographic check that the service is actually running inside a valid TEE and that it runs the expected software. Such checks needs to rely on third parties. The attestation verification is performed by a verifier - that can be either user's computer or a designated verifier as discussed in [I-D.ietf-rats-architecture] and [I-D.ietf-rats-attestation-results] The verifier checks that (a) the cryptographic attestation refers to a server machine that is acceptable to the user (e.g., manufactured by a manufacturer it trusts, CPU features considered secure are used, features considered insecure are turned off, etc.) (b) that the software image designated as being run in the attestation is a software image that the relying party (end user) is willing to use (e.g., has a hash that matches a known software that does not log user actions, or is vouched as trustworthy by another party that the relying party trusts).

## 7. Operational Considerations

This section discusses some aspects of the Confidential Computing arrangement for DNS, based on the authors' experience with these systems.

### 7.1. Operations

Given that the service executes confidentially, and is not observable even by the owner of the hardware, the operations model becomes different. Some different models may be applied:

- \* The service executes on a hardware platform (such as a commercial cloud service) that has no access to information, but there is some other management entity that does have access. The control functions of this entity can communicate with the service instances running in TEEs, and have access to the internal state and statistics of the service instances.
- \* Truly confidential operations where the service and hardware owners have decided to deploy a service that really does not expose private user information to anyone, including themselves.

It is not clear how the first model differs from currently deployed service models. It merely makes it possible to run a service without exposing information to, say, the cloud provider, but any data collection about user behaviours would still be possible for the service owner.

As a result, this document focuses mostly on the second model. For some functions, such as DNS resolution, it is possible to hide all user-related information, and our document argues that we should do so.

Of course, the owners of a service do need some information to run the service, from an efficiency, scaling, problem tracking, and security monitoring point of view. The service operator may even benefit from seeing some overall trend information about various queries and traffic. This does not have to mean exposing individual user behaviours, however.

The authors have worked with aggregate statistics to be able to provide load, performance, memory usage, cache statistics, error, and other information out of the confidential processes. This helps the operator understand the health and status of various service instances. Even with aggregate statistics, there are some danger of revealing private information. For instance, even a sum of counters across all clients can reveal counters associated with an individual user, if the aggregate counters can be sampled at any time with arbitrary precision. For instance, the actions of a single client can be determined by sampling the statistics before and after that client sent a message.

A simplistic approach to producing safer statistics in such cases is to truncate and/or obfuscate the least significant bits of the statistics. It is often necessary to tailor such truncation to the types of measurements, e.g., number of requests is typically a very large number while the number of specific errors is usually small. Truncation could of course be done dynamically. More generally, the set of information provided to the operator about the confidential process could be viewed in light of differential privacy.

Another complementary approach is to provide statistics only at set intervals, or after a sufficient amount of new traffic has been received.

Another complementary technique to monitor the health of confidential services is the use of probes to ensure that the services function correctly. Probes can also measure the performance of the services.

The case of excessive service conditions due to Denial-of-Service attacks is discussed further under the Security Considerations section.

## 7.2. Debugging

Various error conditions and software issues may occur, as is usual with any service. There is a need to monitor problems that occur inside the service or at the client. This can be done, for instance, with the help of various statistics discussed earlier.

Some of the monitored conditions should include:

- \* All major (or preferably even minor) error conditions should have an associated counter. This is necessary as no traditional logging can be reasonably provided that would otherwise have entries for, say, "client IP 203.0.113.0 sent a malformed request". While some errors can be expected at any time, a major increase in specific issues can indicate a problem. As a result, the counters need to be monitored and issues investigated as needed.
- \* Client connection failures, which might indicate software version, trust root or other configuration problems.

Of course, for dedicated software testing purposes (such as debugging interoperability problems), even confidential services need to be run in a mode that exposes everything. Actual clients and users **MUST** be able to ensure that they are connected to a production service instance. This can be done by providing debugging status as part of the remote attestation, so that clients can verify it is off. Alternatively, testing versions of the service are simply not listed as trusted software versions.

## 7.3. Dependencies

The use of Confidential Computing introduces three additional dependencies to the system:

There is a need to be able to verify that the CPU executing the service is a legitimate CPU with the right hardware, and that the software being run for the service is acceptable. While this can be hard coded information in the service clients, in practice there is often a need to rely on other parties for scalability. As a result, there are two dependencies for legitimate CPU verification and for checking acceptable software versions. These are services that need to be run, and/or their use need to be agreed and possibly contracted for. The CPU manufacturer often plays a role in the CPU verification.

The third dependency is on the client. Depending on specific protocol arrangements, Confidential Computing services often can

serve unmodified clients, but for the full benefits and for validating attestations or software images, client changes are necessary. The necessary communications may happen as part of TLS negotiations or other general purpose protocols [I-D.mandyam-tokbind-attest], [I-D.ietf-rats-eat].

#### 7.4. Additional services

Many services employ information that can be used to perform additional services beyond the basic task. For instance, knowledge about what the users requests or who the user is can be used for various optimizations or additional information that can be delivered to the user. Or the user can provide some additional information that is taken into account by the service.

One concern with these types of additional services is that the information used by them can be privacy sensitive. But Confidential Computing can assist in this as well, as long as the relevant information stays only within the TEE, it is better protected than by, e.g., providing that extra information to a regular service on the Internet.

Conversely, care needs to be taken whenever the service needs to relay some information outside the TEE. Some specific situations where this is needed with DNS are discussed in Section 7.1.

One example of additional services is that aggregate, privacy-sensitive data may be produced about trends in a confidentially run service, if it will not be possible to separate individual users from that data. For instance, it would be difficult sell information about individual users to help with targeted advertising, but the overall popularity of some websites could be measured.

#### 7.5. Performance

Confidential Computing technology may impact performance. Nakatsuka et al. [PDoT] report on DNS resolution within a TEE where their solution could outperform the open source Unbound DNS server in certain scenarios, especially in situations where there are not a lot of DNS client connections. We concur their suggestion that at current stage of Confidential Computing technology, possible implementations may be more suited for local DNS resolution services rather than global scale implementation, where the performance hit would be much more significant. Nonetheless with Confidential Computing technology ever evolving we believe the low performance overhead solutions will be possible in foreseeable future.

Other things being equal there's likely some performance hit, as current Confidential Computing technology typically involves separating a server into two parts, the trusted and untrusted parts. In practice, all communications need to go through both, and the communication between the two parts consumes some cycles. There are also current limitations on amount of memory or threads supported by these technologies. However, newer virtualization-based confidential computing TEE approaches are likely going to improve these aspects.

Another performance hit comes from the overhead related to running the attestation process, and passing the necessary extra information in the communications protocols with the clients. In general, this works best when the cost of the setup is amortized over a long-lived session. Such sessions may exist between DoT/DoH-enabled clients and resolvers. Also, there are many possible arrangements and possible parties involved in attestation, see [I-D.ietf-rats-architecture].

## 8. Security Considerations

Security issues in this arrangement are discussed below.

### 8.1. Observations from outside the TEE

While a TEE is considered to be secure and not observable, there may be signs outside the TEE that can reveal information.

For instance, a server may receive a request from a client and immediately send out a question to a server in the Internet about a particular domain name. Observers - such as the owner of the server computer or the cloud farm - may be able to link incoming user queries to outgoing questions

Caching, randomly made other traffic, and timing obfuscation can deter such attacks, at least to an extent.

### 8.2. Trust Relationships

For scaling reasons, the arrangement typically depends on the ability to have trusted parties (a) for attesting the validity of a particular CPU being manufactured by a CPU manufacturer, and (b) for determining whether a particular software image hash is acceptable for the task it is advertising to do.

Such trusted parties need to be configured, which presents an additional operational burden. The information can of course be provided as part of a device manufacturer's or application's initial configuration, or be provided independently similar to how, for instance, certificate authorities are run.

It is important to recognize that mere use of technology is not sufficient to make the system secure. With communications, establishing a secure, encrypted channel is of no use if it is not with the intended party due to a certificate authority that proved to be untrustworthy. With confidential computing, the same applies: one has to have someone who can assert that a CPU is capable of performing the confidential computing task and that the indicated software is good for performing the task that the user expects it to perform. That being said, when such trusted parties can be found, the service performed by the server can become much more privacy friendly.

### 8.3. Denial-of-Service Attacks

To paraphrase an old philosophical question, "If an evil packet is sent behind the veil of encryption and no one is around to lift it, did an attack happen?" [Chautauquan]

Denial-of-Service attacks are a more serious form of the problems with operating services that the operator (intentionally) does not fully see. There needs to be means to deal with these attacks.

Attacks that can be identified by particularly high traffic flows from externally observable sources (e.g., source IP address) can of course still be dealt with in similar ways as we do in more open server designs.

But this is often not enough, and for this purpose some additional support is needed in the systems, for both detection of attacks and reacting to them.

One detection technique is to use the aggregate/truncated statistics to analyze anomalous behaviour. Another technique is to have the confidential part of the service produce extra information about events that cross a threshold. For instance, a particular error may occur exceptionally frequently, say among millions of requests, and this could warrant exposing either something about the request (e.g., the associated domain name) or something about the client (e.g., connection type, protocol details, or sender address).

The operator of the services needs to be able to react to possible attacks as well. One technique is to be able to provide instruction to the confidential part of the service to refuse service for specific requests (e.g., specific domain names) or for specific clients (e.g., coming from specific addresses). Alternatively, the service can also dynamically react to issues, e.g., by starting to reduce the amount of resources dedicated to some classes of requests that for some reason are starting to require exceptionally high

amount of resources. These techniques do not endanger user privacy, but may of course impact provided service.

#### 8.4. Other vulnerabilities

Like all security mechanisms, this solution is not a panacea. It relies on the correct operation of a number of technologies and entities. For instance, CPU bugs or side channel vulnerabilities can cause information leaks to become possible. While confidential computing offers a layer of protection against attacks even from the owner of the computer hardware or the operating system, it is believed that this protection does not extend to sophisticated physical attacks, such being able to study chips with an electron microscope.

And as discussed above, it is also critical to check what software is being run, as otherwise any possible benefit would be negated by the possibly negligent or nefarious actions the unchecked software makes.

The mechanism does offer an additional layer of defense, however. It allows some of the trust that we place on our cloud platform owners, CPUs, and software applications to be verified and controlled with technical means. It may have some remaining vulnerabilities, but we obviously already depend on, for instance, the correct operation of our computing platforms. As such, Confidential Computing works to reduce some of the vulnerabilities in this area.

It should also be a desirable feature for users. A service that offers Confidential Computing-based protection of user data and can show that its software does not leak user-specific information is likely going to be more attractive to users than one that provides no such assurances. Of course, overall user choice depends on many factors beyond privacy, such as cost, ease of use, switching costs, and so on.

There is also a danger of attacks or pressure from intelligence agencies that could result in, e.g., the use of unpublicized vulnerabilities in an attempt to dwarf the protections in Confidential Computing. This could be used to perform pervasive monitoring, for instance [RFC7258]. Even so, it is always beneficial to push the costs and difficulty for attackers. Requiring parties who perform pervasive monitoring to employ complex technical attacks rather than being able to request logs from a service provider significantly increases the difficulty and risk associated with such monitoring.

## 9. Recommendations

Data held by servers SHOULD receive at least as much security attention as communications do.

The authors would like to draw attention to the problem of data leaks, particularly for data in use, and RECOMMEND the application of all available tools to prevent inappropriate access to users' information.

This is particularly crucial for DNS resolution services that have the potential to learn user's browsing histories. But the principles apply also to other services.

While using Confidential Computing without other modifications to the service in question is possible, real benefits can only be realized when the actual service is built for the purpose of avoiding data leaks or user data capture. Systems may need to be tuned or modified, for instance they MUST NOT produce logs that would negate purpose of running them inside a TEE to begin with. Mechanisms SHOULD be found to enable debugging and the detection of fault situations and attacks, again without exposing private information relating to individual users.

Some computing services can proceed on their own and require no interaction with the rest of the world. These are easier to secure. Even then, care SHOULD be taken to avoid request-response timing to provide information useful for side-channel attacks. If so, the owner of the server hardware can not determine much about what was going on.

However, other services may require interaction with other systems, such as is the case with a DNS resolver needing to find out a particular name that is not in a cache or whose cache entry has expired. This is because the resolution service is not a self-contained computation task but ultimately needs, at least in some cases, interaction with the rest of the world.

Consequently, the resolver needs to collaborate with other network nodes that are not even in the same administrative domain and cannot be guaranteed to subscribe to the same principles of protecting user's information. In this case, even if communications to other entities are encrypted, the potentially untrusted party at the other end of the communications may leak information.

In such communications, care SHOULD be taken to avoid exposing any information that would identify users, or allow fingerprinting the capabilities of those users' systems. Similarly, care SHOULD be



taken to avoid exposing any timing information that would allow the owner of the server hardware to determine what is going on, e.g., which users are asking for what names. Even so, vulnerabilities may appear if the attacker can force the system to behave in a particular way, by, e.g., forcing cache overflow, overloading it with traffic it knows about, etc.

The situation is slightly different when the interaction is with other systems that form a part of the same administrative domain. In particular, if those other systems employ similar confidential computing setup, and an encrypted channel is used, then some additional security can be provided compared to communicating with other entities in the Internet.

## 10. Acknowledgments

The authors would like to thank Juhani Kauppi, Jimmy Kjaellman, and Tero Kauppinen for their work on systems supporting some of the ideas discussed in this memo, and Dave Thaler, Daniel Migault, Karl Norrman, and Christian Schaefer for significant feedback on early version of this draft. The author would also like to thank Marcus Ihlar, Maria Luisa Mas, Miguel Angel Munos De La Torre Alonso, Jukka Ylitalo, Bengt Sahlin, Tomas Mecklin, Ben Smeets and many others for interesting discussions in this problem space.

## 11. References

### 11.1. Normative References

- [RFC1035] Mockapetris, P.V., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

### 11.2. Informative References

- [AMD] Kaplan, D., Powell, J., and T. Woller, "AMD Memory Encryption", AMD White Paper , April 2016.

## [Cambridge]

Isaak, J. and M. Hanna, "User Data Privacy: Facebook, Cambridge Analytica, and Privacy Protection", Computer 51.8 (2018): 56-59, <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8436400> , 2018.

## [CC]

Rashid, F.Y., "What Is Confidential Computing?", IEEE Spectrum, <https://spectrum.ieee.org/computing/hardware/what-is-confidential-computing> , May 2020.

## [CCC-Deepdive]

Confidential Computing Consortium, ., "A Technical Analysis of Confidential Computing", <https://confidentialcomputing.io/whitepaper-02-latest> , January 2021.

## [Chautauquan]

"The Chautauquan", Volume 3, Issue 9, p. 543 , June 1883.

## [Comparison]

Mofrad, S., Zhang, F., Lu, S., and W. Shi, "A comparison study of intel SGX and AMD memory encryption technology", HASP '18, Proceedings of the 7th International Workshop on Hardware and Architectural Support for Security and Privacy, Pages 1-8, <https://doi.org/10.1145/3214292.3214301> , June 2018.

## [Digging]

Hammouchi, H., Cherqi, O., Mezzour, G., Ghogho, M., and M. El Koutbi, "Digging Deeper into Data Breaches: An Exploratory Data Analysis of Hacking Breaches Over Time", Procedia Computer Science, Volume 151, pp. 1004-1009, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2019.04.141>, <https://www.sciencedirect.com/science/article/pii/S1877050919306064> , 2019.

## [Efficient]

Suh, G.E., Clarke, D., Gasend, B., van Dijk, M., and S. Devadas, "Efficient memory integrity verification and encryption for secure processors", Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-36, San Diego, CA, USA, pp. 339-350, doi: 10.1109/MICRO.2003.1253207 , 2003.

## [I-D.arkko-arch-infrastructure-centralisation]

Arkko, J., "Centralised Architectures in Internet Infrastructure", Work in Progress, Internet-Draft, draft-arkko-arch-infrastructure-centralisation-00, 4 November

2019, <<https://www.ietf.org/archive/id/draft-arkko-arch-infrastructure-centralisation-00.txt>>.

[I-D.arkko-farrell-arch-model-t-redux]

Arkko, J. and S. Farrell, "Internet Threat Model Evolution: Background and Principles", Work in Progress, Internet-Draft, draft-arkko-farrell-arch-model-t-redux-01, 22 February 2021, <<https://www.ietf.org/archive/id/draft-arkko-farrell-arch-model-t-redux-01.txt>>.

[I-D.iab-dedr-report]

Arkko, J. and T. Hardie, "Report from the IAB Workshop on Design Expectations vs. Deployment Reality in Protocol Development", Work in Progress, Internet-Draft, draft-iab-dedr-report-01, 2 November 2020, <<https://www.ietf.org/archive/id/draft-iab-dedr-report-01.txt>>.

[I-D.ietf-dprive-dnssoquic]

Huitema, C., Mankin, A., and S. Dickinson, "Specification of DNS over Dedicated QUIC Connections", Work in Progress, Internet-Draft, draft-ietf-dprive-dnssoquic-02, 22 February 2021, <<https://www.ietf.org/archive/id/draft-ietf-dprive-dnssoquic-02.txt>>.

[I-D.ietf-rats-architecture]

Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote Attestation Procedures Architecture", Work in Progress, Internet-Draft, draft-ietf-rats-architecture-12, 23 April 2021, <<https://www.ietf.org/archive/id/draft-ietf-rats-architecture-12.txt>>.

[I-D.ietf-rats-eat]

Mandyam, G., Lundblade, L., Ballesteros, M., and J. O'Donoghue, "The Entity Attestation Token (EAT)", Work in Progress, Internet-Draft, draft-ietf-rats-eat-10, 7 June 2021, <<https://www.ietf.org/archive/id/draft-ietf-rats-eat-10.txt>>.

[I-D.ietf-tls-esni]

Rescorla, E., Oku, K., Sullivan, N., and C. A. Wood, "TLS Encrypted Client Hello", Work in Progress, Internet-Draft, draft-ietf-tls-esni-11, 14 June 2021, <<https://www.ietf.org/archive/id/draft-ietf-tls-esni-11.txt>>.

[I-D.lazanski-smart-users-internet]

Lazanski, D., "An Internet for Users Again", Work in

Progress, Internet-Draft, draft-lazanski-smart-users-internet-00, 8 July 2019,  
<<https://www.ietf.org/archive/id/draft-lazanski-smart-users-internet-00.txt>>.

[I-D.mandyam-tokbind-attest]

Mandyam, G., Lundblade, L., and J. Azen, "Attested TLS Token Binding", Work in Progress, Internet-Draft, draft-mandyam-tokbind-attest-07, 24 January 2019,  
<<https://www.ietf.org/archive/id/draft-mandyam-tokbind-attest-07.txt>>.

[I-D.reddy-add-server-policy-selection]

Reddy, T., Wing, D., Richardson, M. C., and M. Boucadair, "DNS Server Selection: DNS Server Information with Assertion Token", Work in Progress, Internet-Draft, draft-reddy-add-server-policy-selection-08, 29 March 2021,  
<<https://www.ietf.org/archive/id/draft-reddy-add-server-policy-selection-08.txt>>.

[I-D.thomson-tmi]

Thomson, M., "Principles for the Involvement of Intermediaries in Internet Protocols", Work in Progress, Internet-Draft, draft-thomson-tmi-01, 3 January 2021,  
<<https://www.ietf.org/archive/id/draft-thomson-tmi-01.txt>>.

[I-D.voit-rats-attestation-results]

Voit, E., Birkholz, H., Hardjono, T., Fossati, T., and V. Scarlata, "Attestation Results for Secure Interactions", Work in Progress, Internet-Draft, draft-voit-rats-attestation-results-01, 10 June 2021,  
<<https://www.ietf.org/archive/id/draft-voit-rats-attestation-results-01.txt>>.

[Innovative]

Ittai, A., Gueron, S., Johnson, S., and V. Scarlata, "Innovative Technology for CPU Based Attestation and Sealing", HASP'2013 , 2013.

[Mem]

Henson, M. and S. Taylor, "Memory encryption: a survey of existing techniques", ACM Computing Surveys volume 46 issue 4 , 2014.

[MozTRR]

Mozilla, ., "Security/DOH-resolver-policy", <https://wiki.mozilla.org/Security/DOH-resolver-policy> , 2019.

- [PDoT] Nakatsuka, Y., Paverd, A., and G. Tsudik, "PDoT: Private DNS-over-TLS with TEE Support", Digit. Threat.: Res. Pract., Vol. 2, No. 1, Article 3, <https://dl.acm.org/doi/fullHtml/10.1145/3431171> , February 2021.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <<https://www.rfc-editor.org/info/rfc3552>>.
- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <<https://www.rfc-editor.org/info/rfc7258>>.
- [RFC7626] Bortzmeyer, S., "DNS Privacy Considerations", RFC 7626, DOI 10.17487/RFC7626, August 2015, <<https://www.rfc-editor.org/info/rfc7626>>.
- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [RFC8324] Klensin, J., "DNS Privacy, Authorization, Special Uses, Encoding, Characters, Matching, and Root Structure: Time for Another Look?", RFC 8324, DOI 10.17487/RFC8324, February 2018, <<https://www.rfc-editor.org/info/rfc8324>>.
- [RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/info/rfc8484>>.
- [RFC8558] Hardie, T., Ed., "Transport Protocol Path Signals", RFC 8558, DOI 10.17487/RFC8558, April 2019, <<https://www.rfc-editor.org/info/rfc8558>>.
- [RFC8932] Dickinson, S., Overeinder, B., van Rijswijk-Deij, R., and A. Mankin, "Recommendations for DNS Privacy Service Operators", BCP 232, RFC 8932, DOI 10.17487/RFC8932, October 2020, <<https://www.rfc-editor.org/info/rfc8932>>.
- [SGX] Hoekstra, M.E., "Intel(R) SGX for Dummies (Intel(R) SGX Design Objectives)", Intel, <https://software.intel.com/content/www/us/en/develop/blogs/protecting-application-secrets-with-intel-sgx.html> , September 2013.

- [SmartTV] Malkin, N., Bernd, J., Johnson, M., and S. Egelman, "What Can't Data Be Used For? Privacy Expectations about Smart TVs in the U.S.", European Workshop on Usable Security (Euro USEC), [https://www.ndss-symposium.org/wp-content/uploads/2018/06/eurousec2018\\_16\\_Malkin\\_paper.pdf](https://www.ndss-symposium.org/wp-content/uploads/2018/06/eurousec2018_16_Malkin_paper.pdf) , 2018.
- [Stallman] Stallman, R., "Can You Trust Your Computer?", GNU.org, <https://www.gnu.org/philosophy/can-you-trust.html> , n.d..
- [Toys] Chu, G., Apthorpe, N., and N. Feamster, "Security and Privacy Analyses of Internet of Things Childrens' Toys", IEEE Internet of Things Journal 6.1 (2019): 978-985, <https://arxiv.org/pdf/1805.02751.pdf> , 2019.
- [Unread] Obar, J. and A. Oeldorf, "The biggest lie on the internet{:} Ignoring the privacy policies and terms of service policies of social networking services", Information, Communication and Society (2018): 1-20 , 2018.
- [Vastaamo] Redcross Finland, ., "Read this if your personal data was leaked in the Vastaamo data system break-in", <https://www.redcross.fi/news/20201029/read-if-your-personal-data-was-leaked-vastaamo-data-system-break> , October 2020.

#### Authors' Addresses

Jari Arkko  
Ericsson

Email: [jari.arkko@ericsson.com](mailto:jari.arkko@ericsson.com)

Jiri Novotny  
Ericsson

Email: [jiri.novotny@ericsson.com](mailto:jiri.novotny@ericsson.com)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 14 July 2022

M. Finkel  
The Tor Project  
B. Lassey  
Google  
L. Iannone  
Huawei  
J.B. Chen  
Google  
10 January 2022

IP Address Privacy Considerations  
draft-ip-address-privacy-considerations-03

Abstract

This document provides an overview of privacy considerations related to user IP addresses. It includes an analysis of some current use cases for tracking of user IP addresses, mainly in the context of anti-abuse. It discusses the privacy issues associated with such tracking and provides input on mechanisms to improve the privacy of this existing model. It then captures requirements for proposed 'replacement signals' for IP addresses from this analysis. In addition, existing and under-development techniques are evaluated for fulfilling these requirements.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the mailing list (), which is archived at .

Source for this draft and an issue tracker can be found at <https://github.com/ShivanKaul/draft-ip-address-privacy>.

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 14 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 . . . . .	3
2. Terminology . . . . .	4
2.1. Categories of Interaction . . . . .	4
3. IP address tracking . . . . .	5
3.1. IP address use cases . . . . .	5
3.1.1. Anti-abuse . . . . .	5
3.1.2. DDoS and Botnets . . . . .	5
3.1.3. Multi-platform threat models . . . . .	6
3.1.4. Rough Geolocation . . . . .	6
3.2. Implications of IP addresses . . . . .	7
3.2.1. Next-User Implications . . . . .	7
3.2.2. Privacy Implications . . . . .	7
3.3. IP Privacy Protection and Law . . . . .	8
3.4. Mitigations for IP address tracking . . . . .	8
4. Replacement signals for IP addresses . . . . .	9
4.1. Signals . . . . .	9
4.1.1. Adoption . . . . .	10
4.1.2. Privacy Considerations . . . . .	11
4.1.3. Provenance . . . . .	12
4.1.4. Applying Appropriate Signals . . . . .	12
4.2. Evaluation of existing technologies . . . . .	13
5. Security Considerations . . . . .	14
6. IANA Considerations . . . . .	14
7. References . . . . .	14
7.1. Normative References . . . . .	14



7.2. Informative References . . . . .	14
Acknowledgments . . . . .	16
Authors' Addresses . . . . .	16

## 1. Introduction

The initial intention of this draft is to capture an overview of the problem space and research on proposed solutions concerning privacy considerations related to user IP addresses (informally, IP privacy). The draft is likely to evolve significantly over time and may well split into multiple drafts as content is added.

Tracking of IP addresses is common place on the Internet today, and is particularly widely used in the context of anti-abuse, e.g. anti-fraud, DDoS management, and child protection activities. IP addresses are currently used in determining "reputation" [RFC5782] in conjunction with other signals to protect against malicious traffic, since these addresses are usually a relatively stable identifier of a request's origin. Servers use these reputations in determining whether or not a given packet, connection, or flow likely corresponds to malicious traffic. In addition, IP addresses are used in investigating past events and attributing responsibility.

However, identifying the activity of users based on IP addresses has clear privacy implications ([WEBTRACKING1], [WEBTRACKING2]), e.g. user fingerprinting and cross-site identity linking. Many technologies exist today that allow users to obfuscate their external IP address to avoid such tracking, e.g. VPNs ([VPNCMP1], [VPNCMP2]) and Tor ([TOR], [VPNTOR]). Several new technologies are emerging, as well, in the landscape, e.g. Apple iCloud Private Relay [APPLEPRIV], Gnatcatcher [GNATCATCHER], and Oblivious technologies (ODoH [I-D.paully-dprive-oblivious-doh], OHTTP [I-D.thomson-ohai-ohttp]).

General consideration about privacy for Internet protocols can be found in [RFC6973]. This document builds upon [RFC6973] and more specifically attempts to capture the following aspects of the tension between valid use cases for user identification and the related privacy concerns, including:

- \* An analysis of the current use cases, attempting to categorize/group such use cases where commonalities exist.
- \* Find ways to enhance the privacy of existing uses of IP addresses.
- \* Generating requirements for proposed 'replacement signals' from this analysis (these could be different for each category/group of use cases).

- \* Research to evaluate existing technologies or propose new mechanisms for such signals.

With the goal of replacing IP addresses as a fundamental signal, the following sections enumerate existing use cases and describe applicable substitution signals. This description may not be exhaustive due to the breadth of IP address usage.

## 2. Terminology

(Work in progress)

This section defines basic terms used in this document, with references to pre-existing definitions as appropriate. As in [RFC4949] and [RFC6973], each entry is preceded by a dollar sign (\$) and a space for automated searching.

- \* \$ Identity: Extending [RFC6973], an individual's attributes may only identify an individual up to an anonymity set within a given context.
- \* \$ Reputation: A random variable with some distribution. A reputation can either be "bad" or "good" with some probability according to the distribution.
- \* \$ Reputation context: The context in which a given reputation applies.
- \* \$ Reputation proof: A non-interactive zero knowledge proof of a reputation signal.
- \* \$ Reputation signal: A representative of a reputation.
- \* \$ Service provider: An entity that provides a service on the Internet; examples services include hosted e-mail, e-commerce sites, and cloud computing platforms.

### 2.1. Categories of Interaction

Interactions between parties on the Internet may be classified into one (or more) of three categories:

- \* \$ Private Interaction: An interaction occurring between mutually consenting parties, with a mutual expectation of privacy.
- \* \$ Public Interaction: An interaction occurring between multiple parties that are not engaged in a Private Interaction.

- \* **\$ Consumption:** An interaction where one party primarily receives information from other parties.

### 3. IP address tracking

#### 3.1. IP address use cases

##### 3.1.1. Anti-abuse

IP addresses are a passive identifier used in defensive operations. They allow correlating requests, attribution, and recognizing numerous attacks, including:

- \* account takeover
- \* advertising fraud (e.g., click-fraud)
- \* disinformation operations (e.g., detecting scaled and/or coordinated attacks)
- \* financial fraud (e.g., stolen credit cards, email account compromise)
- \* malware/ransomware (e.g., detecting C2 connections)
- \* phishing
- \* real-world harm (e.g., child abuse)
- \* scraping (e.g., e-commerce, search)
- \* spam (e.g., email, comments)
- \* vulnerability exploitation (e.g., "hacking")

Malicious activity recognized by one service provider may be shared with other services [RFC5782] as a way of limiting harm.

##### 3.1.2. DDoS and Botnets

Cyber-attackers can leverage the good reputation of an IP address to carry out specific attacks that wouldn't work otherwise. Main examples are Distributed Denial of Service (DDoS) attacks carried out by spoofing a trusted (i.e., having good reputation) IP address (which may or may not be the victim of the attack) so that the servers used to generate the DDoS traffic actually respond to the attackers trigger (i.e., spoofed packets). Similarly botnets may use spoofed addresses in order to gain access and attack services that

otherwise would not be reachable.

#### 3.1.3. Multi-platform threat models

As siloed (single-platform) abuse defenses improve, abusers have moved to multi-platform threat models. For example, a public discussion platform with a culture of anonymity may redirect traffic to YouTube as a video library, bypassing YouTube defenses that otherwise reduce exposure of potentially harmful content. Similarly, a minor could be solicited by an adult impersonating a child on a popular social media platform, then redirected to a smaller, less established and less defended platform where illegal activity could occur. Phishing attacks are also common. There are many such cross-platform abuse models and they cause significant public harm. IP addresses are commonly used to investigate, understand and communicate these cross-platform threats. There are very few alternatives for cross-platform signals.

#### 3.1.4. Rough Geolocation

A rough geolocation can be inferred from a client's IP address, which is commonly known as either IP-Geo or Geo-IP. This information can have several useful implications. When abuse extends beyond attacks in the digital space, IP addresses may help identify the physical location of real-world harm, such as child exploitation.

##### 3.1.4.1. Legal compliance

Legal and regulatory compliance often needs to take the jurisdiction of the client into account. This is especially important in cases where regulations are mutually contradictory (i.e. there is no way to be in legal compliance universally). Because Geo-IP is often bound to the IP addresses a given ISP uses, and ISPs tend to operate within national borders, Geo-IP tends to be a good fit for server operators to comply with local laws and regulations

##### 3.1.4.2. Contractual obligations

Similar to legal compliance, some content and media has licensing terms that are valid only for certain locations. The rough geolocation derived from IP addresses allow this content to be hosted on the web.

#### 3.1.4.3. Locally relevant content

Rough geolocation can also be useful to tailor content to the client's location simply to improve their experience. A search for "coffee shop" can include results of coffee shops within reasonable travel distance from a user rather than generic information about coffee shops, a merchant's website could show brick and mortar stores near the user and a news site can surface locally relevant news stories that wouldn't be as interesting to visitors from other locations.

### 3.2. Implications of IP addresses

#### 3.2.1. Next-User Implications

When an attacker uses IP addresses with "good" reputations, the collateral damage poses a serious risk to legitimate service providers, developers, and end users. IP addresses may become associated with a "bad" reputation from temporal abuse, and legitimate users may be affected by blocklists as a result. This unintended impact may hurt the reputation of a service or an end user [RFC6269].

#### 3.2.2. Privacy Implications

IP addresses are sent in the clear throughout the packet journey over the Internet. As such, any observer along the path can pick it up and use it for various tracking purposes. Beside basic information about the network or the device, it is possible to associate an IP address to an end user, hence, the relevance of IP addresses for user privacy. A very short list of information about user, device, and network that can be obtained via the IP address.

- \* Determine who owns and operates the network. Searching the WHOIS database using an IP address can provide a range of information about the organization to which the address is assigned, including a name, phone number, and civic address;
- \* Through a reverse DNS lookup and/or traceroute the computer name can be obtained, which often contains clues to logical and physical location;
- \* Geo-localisation of the device (hence the user) through various techniques [GEOIP]. Depending on the lookup tool used, this could include country, region/state, city, latitude/longitude, telephone area code and a location-specific map;

- \* Search the Internet using the IP address or computer names. The results of these searches might reveal peer-to-peer (P2P) activities (e.g., file sharing), records in web server log files, or glimpses of the individual's web activities (e.g., Wikipedia edits). These bits of individuals' online history may reveal their political inclinations, state of health, sexuality, religious sentiments and a range of other personal characteristics, preoccupations and individual interests;
- \* Seek information on any e-mail addresses used from a particular IP address which, in turn, could be the subject of further requests for subscriber information.

### 3.3. IP Privacy Protection and Law

This section aim at providing some basic information about main example of laws adopted worldwide and related to IP address privacy (usually these laws area by product of the broader user privacy protection).

Possible content (to focus only on technical IP address related aspects):

- \* GDPR (General Data Protection Regulation) - EUROPE: Europe considers IP addresses as personal identification information that should be treated like any other personal information e.g. social security number.
- \* The United States has opted for a different approach to data protection. Instead of formulating one all-encompassing regulation such as the EU's GDPR, the US chose to implement sector-specific privacy and data protection regulations that work together with state laws to safeguard American citizens' data.
- \* In 2020, China released the first draft of Personal Information Protection Law (PIPL). The PIPL is the equivalent of European GDPR and will have significant influence.
- \* Japan Protection of Personal Information (APPI) Act (recent changes put the act close to the GDPR model).

### 3.4. Mitigations for IP address tracking

The ability to track individual people by IP address has been well understood for decades. Commercial VPNs and Tor are the most common methods of mitigating IP address-based tracking.

- \* Commerical VPNs offer a layer of indirection between the user and the destination, however if the VPN endpoint's IP address is static then this simply substitutes one address for another. In addition, commerical VPNs replace tracking across sites with a single company that may track their users' activities.
- \* Tor is another mitigation option due to its dynamic path selection and distributed network of relays, however its current design suffers from degraded performance. In addition, correct application integration is difficult and not common.
- \* Address anonymization (e.g. [GNATCATCHER] and similar):
  - [GNATCATCHER] is a single-hop proxy system providing more protection against third-party tracking than a traditional commercial VPN. However, its design maintains the industry-standard reliance on IP addresses for anti-abuse purposes and it provides near backwards compatibility for select services that submit to periodic audits.
  - [APPLEPRIV] iCloud Private Relay is described as using two proxies between the client and server, and it would provide a level of protection somewhere between a commercial VPN and Tor.
- \* Recent interest has resulted in new protocols such as Oblivious DNS (ODoH ([I-D.pauly-oblivious-doh-02.html](https://www.ietf.org/archive/id/draft-pauly-oblivious-doh-02.html)))) and Oblivious HTTP (OHTTP ([I-D.thomson-http-oblivious](https://www.ietf.org/archive/id/draft-thomson-http-oblivious.html)))). While they both prevent tracking by individual parties, they are not intended for the general-purpose web browsing use case.
- \* Temporary addresses

#### 4. Replacement signals for IP addresses

Fundamentally, the current ecosystem operates by making the immediate peer of a connection accountable for bad traffic, rather than the source of the traffic itself. This is problematic because in some network architectures the peer node of the connection is simply routing traffic for other clients, and any client's use of that node may be only temporary. Ideally, clients could present appropriate identification end-to-end that is separate from the IP address, and uniquely bound to a given connection.

##### 4.1. Signals

There are 7 classes of signals identified in this document that may be used in place of IP addresses. A signal's provenance is a critical property and will be discussed in Section 4.1.3.

- \* ADDRESS\_ESCROW: Provides sufficient information for retroactively obtaining a client's IP address.
- \* IDENTITY\_TRANSPARENCY: Reveals a person's identity within a context.
- \* IS\_HUMAN: Informs the recipient that, most likely, a human recently proved their presence on the opposite end of the connection.
- \* PEER\_INTEGRITY: Provides a secure, remote attestation of hardware and/or software state.
- \* REIDENTIFICATION: Provides a mechanism for identifying the same user across different connections within a time period.
- \* REPUTATION: Provides the recipient with a proof of reputation from a reputation provider.
- \* SOURCE\_ASN: Reveals the ASN from which the client is connecting.

In some situations one of the above signals may be a sufficient replacement signal in isolation, or more than one signal may be needed in combination.

Separately, there are three signal categories that are out-of-scope for this document but are important improvements for mitigating abuse on platforms.

- \* publisher norms: Standard expectations of publishers including identity transparency and conflicts of interest.
- \* protocol improvements: Increasing security of existing protocols.
- \* ecosystem improvements: Reducing reliance on less secure systems, for example, migrating user authentication from password-based to WebAuthn [WEBAUTHN] and relying on multiple factors (MFA).

#### 4.1.1. Adoption

Adoption of replacement signals requires coordination between user agents, service providers, and proxy services. Some user agents and proxy services may support only a subset of these signals, while service providers may require additional signals. A mechanism of negotiation may be needed for communicating these requirements.



In addition, service providers should only require a signal within the scope it will be used. In the same way that service providers only require user authentication when the user requests access to a non-public resource, a signal should not be pre-emptively requested before it is needed. The categories of interaction described above may help define scopes within a service, and they may help communicate to the user the reasoning for requiring a signal.

#### 4.1.2. Privacy Considerations

A signal should not be required without clear justification, service providers should practice data minimization [RFC6973] wherever possible. Requiring excessive signals may be more harmful to user privacy than requiring IP address transparency. This section provides a more details analysis of some signals.

ADDRESS\_ESCROW gives service providers a time period within which they may obtain the client's IP address, but the information-in-escrow is not immediately available. Service providers should not gain access to the information in secret. A service provider may misuse the information-in-escrow for tracking and privacy-invasion purposes.

PEER\_INTEGRITY partitions users into two groups with valid and invalid hardware/software state, at a minimum. If the signal reveals more information, then it may allow more granular tracking of small sets of devices.

IDENTITY\_TRANSPARENCY may expose significant information about a user to a service provider; the resulting privacy invasion may be significantly worse than IP address transparency causes.

IS\_HUMAN depends on the mechanism used for proving humanness.

REIDENTIFICATION explicitly allows a service provider to associate requests across unlinkable connections. This signal allows for profiling user behavior and tracking user activity without requesting more identifying information. First-party reidentification is a use case for this signal.

REPUTATION partitions users into a set based on their reputation. The privacy invasion associated with this signal is intentionally small.

SOURCE\_ASN allows for identifying request patterns originating from an ASN without providing IP address transparency. However, ASNs are not guaranteed to serve large populations, therefore revealing the source ASN of a request may reveal more information about the user than intended.

#### 4.1.3. Provenance

Replacement signals are only useful if they are trustworthy.

[[OPEN ISSUE: <https://github.com/ShivanKaul/draft-ip-address-privacy/issues/24>]]

#### 4.1.4. Applying Appropriate Signals

As previously discussed, IP addresses are used for various reasons; therefore, describing a one-size-fits-all replacement signal is not appropriate. In addition, the quality and quantity of replacement signals needed by a service depends on the category of interaction of its users and potential attacks on the service.

As an example, the attacks listed above in Section 3.1.1 can be organized into six groups based on the signals that may sufficiently replace IP addresses:

1. IS\_HUMAN, REPUTATION, REIDENTIFICATION, PEER\_INTEGRITY
  - \* advertising fraud (e.g., click-fraud)
  - \* phishing
  - \* scraping (e.g., e-commerce, search)
  - \* spam (e.g., email, comments)
2. IS\_HUMAN, REPUTATION, REIDENTIFICATION, ecosystem improvements
  - \* account takeover
3. IS\_HUMAN, REPUTATION, SOURCE\_ASN
  - \* influence (e.g., brigading, astroturfing)
4. publisher norms, (publisher) IDENTITY\_TRANSPARENCY, PEER\_INTEGRITY
  - \* disinformation operations (e.g., detecting scaled and/or coordinated attacks)

5. publisher norms, (publisher) IDENTITY\_TRANSPARENCY, ADDRESS\_ESCROW
  - \* real-world harm (e.g., child abuse)
6. IDENTITY\_TRANSPARENCY, protocol improvements
  - \* financial fraud (e.g., stolen credit cards, email account compromise)

The remaining two attack categories fall outside of the scope of this document.

- \* malware/ransomware (e.g., detecting C2 connections)
- \* vulnerability exploitation (e.g., "hacking")

Note, IP addresses do not provide a perfect signal in their existing usage, and the above replacement signals do not provide a better signal in all cases.

#### 4.2. Evaluation of existing technologies

Technologies exist that are designed to solve some of the problems described in this document.

Privacy Pass [I-D.ietf-privacypass-protocol] is a useful building block for solving numerous problems. Its design involves an interaction between a client and server where, at the end, the client is issued a set of anonymous tokens. These tokens may be redeemed at a later time, and this redemption should not be linkable with the initial issuance interaction. One existing use case is substituting a CAPTCHA challenge with a token, where successfully solving a CAPTCHA challenge results in a client being issued a set of anonymous tokens, and these tokens may be used in the future to bypass solving another CAPTCHA challenge. Therefore, Privacy Pass may be acceptable as an IS\_HUMAN signal by some service providers. The current token design can't carry additional metadata like a user's reputation or an expiration date, and the tokens are not bound to an identity. The unlinkability property of the tokens is dependent on the implementation of key consistency [I-D.wood-key-consistency].

Trust Token [TRUSTTOKEN] is an extension of Privacy Pass where the issuance and redemption functionality are provided in the browser setting. The tokens are allowed to carry public and private metadata as extensions.

Private Access Tokens [I-D.private-access-tokens] provide a technique for partitioning clients based on a per-origin policy within a time period. Its use cases include rate-limiting access to content and geo-location. PATs could be used as a REIDENTIFICATION signal or a replacement signal for GeoIP, depending on requirements.

## 5. Security Considerations

This draft discussses IP address use cases, underlying requirements, and possible replacement signals. Adoption challenges and privacy considerations for those signals are also discussed. Further work is needed to build and evaluate these signals as suitable replacements for IP addresses.

## 6. IANA Considerations

This document has no IANA actions.

## 7. References

### 7.1. Normative References

- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <<https://www.rfc-editor.org/rfc/rfc4949>>.
- [RFC5782] Levine, J., "DNS Blacklists and Whitelists", RFC 5782, DOI 10.17487/RFC5782, February 2010, <<https://www.rfc-editor.org/rfc/rfc5782>>.
- [RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, DOI 10.17487/RFC6269, June 2011, <<https://www.rfc-editor.org/rfc/rfc6269>>.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <<https://www.rfc-editor.org/rfc/rfc6973>>.

### 7.2. Informative References

- [APPLEPRIV] "Apple iCloud Private Relay", n.d., <<https://appleinsider.com/articles/21/06/10/how-apple-icloud-private-relay-works>>.

- [GEOIP] Dan, O., Parikh, V., and B. Davison, "IP Geolocation Using Traceroute Location Propagation and IP Range Location Interpolation", Companion Proceedings of the Web Conference 2021, DOI 10.1145/3442442.3451888, April 2021, <<https://doi.org/10.1145/3442442.3451888>>.
- [GNATCATCHER] "Global Network Address Translation Combined with Audited and Trusted CDN or HTTP-Proxy Eliminating Reidentification", n.d., <<https://github.com/bslassey/ip-blindness>>.
- [I-D.ietf-privacypass-protocol] Celi, S., Davidson, A., and A. Faz-Hernandez, "Privacy Pass Protocol Specification", Work in Progress, Internet-Draft, draft-ietf-privacypass-protocol-01, 22 February 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-protocol-01>>.
- [I-D.pauly-dprive-oblivious-doh] Kinnear, E., McManus, P., Pauly, T., Verma, T., and C. A. Wood, "Oblivious DNS Over HTTPS", Work in Progress, Internet-Draft, draft-pauly-dprive-oblivious-doh-09, 5 January 2022, <<https://datatracker.ietf.org/doc/html/draft-pauly-dprive-oblivious-doh-09>>.
- [I-D.private-access-tokens] Hendrickson, S., Iyengar, J., Pauly, T., Valdez, S., and C. A. Wood, "Private Access Tokens", Work in Progress, Internet-Draft, draft-private-access-tokens-01, 25 October 2021, <<https://datatracker.ietf.org/doc/html/draft-private-access-tokens-01>>.
- [I-D.thomson-ohai-ohttp] Thomson, M. and C. A. Wood, "Oblivious HTTP", Work in Progress, Internet-Draft, draft-thomson-ohai-ohttp-00, 25 October 2021, <<https://datatracker.ietf.org/doc/html/draft-thomson-ohai-ohttp-00>>.
- [I-D.wood-key-consistency] Davidson, A., Finkel, M., Thomson, M., and C. A. Wood, "Key Consistency and Discovery", Work in Progress, Internet-Draft, draft-wood-key-consistency-01, 19 August 2021, <<https://datatracker.ietf.org/doc/html/draft-wood-key-consistency-01>>.
- [TOR] "The Tor Project", n.d., <<https://www.torproject.org/>>.

## [TRUSTOKEN]

"Trust Token API Explainer", n.d.,  
<<https://github.com/WICG/trust-token-api>>.

## [VPNCMP1]

Osswald, L., Haeberle, M., and M. Menth, "Performance Comparison of VPN Solutions", Universität Tübingen article, DOI 10.15496/PUBLIKATION-41810, May 2020, <<https://doi.org/10.15496/PUBLIKATION-41810>>.

## [VPNCMP2]

Khanvilkar, S. and A. Khokhar, "Virtual private networks: an overview with performance evaluation", IEEE Communications Magazine Vol. 42, pp. 146-154, DOI 10.1109/mcom.2004.1341273, October 2004, <<https://doi.org/10.1109/mcom.2004.1341273>>.

## [VPNTOR]

Ramadhani, E., "Anonymity communication VPN and Tor: A comparative study", n.d., <Journal of Physics Conference Series>.

## [WEBAUTHN]

"Web Authentication: An API for accessing Public Key Credentials Level 2", n.d., <<https://www.w3.org/TR/webauthn-2/>>.

## [WEBTRACKING1]

Bujlow, T., Carela-Espanol, V., Lee, B., and P. Barlet-Ros, "A Survey on Web Tracking: Mechanisms, Implications, and Defenses", Proceedings of the IEEE Vol. 105, pp. 1476-1510, DOI 10.1109/jproc.2016.2637878, August 2017, <<https://doi.org/10.1109/jproc.2016.2637878>>.

## [WEBTRACKING2]

Mishra, V., Laperdrix, P., Vastel, A., Rudametkin, W., Rouvoy, R., and M. Lopatka, "Dont Count Me Out: On the Relevance of IP Address in the Tracking Ecosystem", Proceedings of The Web Conference 2020, DOI 10.1145/3366423.3380161, April 2020, <<https://doi.org/10.1145/3366423.3380161>>.

## Acknowledgments

[[OPEN ISSUE: TODO]]

## Authors' Addresses

Matthew Finkel  
The Tor Project

Email: [sysrq@torproject.org](mailto:sysrq@torproject.org)

Bradford Lassey  
Google

Email: lassey@chromium.org

Luigi Iannone  
Huawei Technologies France S.A.S.U

Email: luigi.iannone@huawei.com

J. Bradley Chen  
Google

Email: bradchen@google.com

Internet Area Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 7 September 2022

Y. Jia  
D. Trossen  
L. Iannone  
Huawei  
P. Mendes  
Airbus  
N. Shenoy  
R.I.T.  
L. Toutain  
IMT-Atlantique  
A. Y. Chen  
Avinta  
D. Farinacci  
lispers.net  
6 March 2022

Gap Analysis in Internet Addressing  
draft-jia-intarea-internet-addressing-gap-analysis-02

Abstract

There exist many extensions to Internet addressing, as it is defined in [RFC0791] for IPv4 and [RFC8200] for IPv6, respectively. Those extensions have been developed to fill gaps in capabilities beyond the basic properties of Internet addressing. This document outlines those properties as a baseline against which the extensions are categorized in terms of methodology used to fill the gap together with examples of solutions doing so.

While introducing such extensions, we outline the issues we see with those extensions. This ultimately leads to consider whether or not a more consistent approach to tackling the identified gaps, beyond point-wise extensions as done so far, would be beneficial. The benefits are the ones detailed in the companion document [I-D.jia-intarea-scenarios-problems-addressing], where, leveraging on the gaps identified in this memo and scenarios provided in [I-D.jia-intarea-scenarios-problems-addressing], a clear problem statement is provided.

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 7 September 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 . . . . .	3
2. Properties of Internet Addressing . . . . .	4
2.1. Property 1: Fixed Address Length . . . . .	4
2.2. Property 2: Ambiguous Address Semantic . . . . .	4
2.3. Property 3: Limited Address Semantic Support . . . . .	5
3. Filling Gaps through Extensions to Internet Addressing Properties . . . . .	5
3.1. Length Extensions . . . . .	5
3.1.1. Shorter Address Length . . . . .	6
3.1.2. Longer Address Length . . . . .	8
3.1.3. Summary . . . . .	10
3.2. Identity Extensions . . . . .	10
3.2.1. Anonymous Address Identity . . . . .	11
3.2.2. Authenticated Address Identity . . . . .	14
3.2.3. Summary . . . . .	15
3.3. Semantic Extensions . . . . .	16
3.3.1. Utilizing Extended Address Semantics . . . . .	17
3.3.2. Utilizing Existing or Extended Header Semantics . . . . .	20
3.3.3. Summary . . . . .	23
4. Overview of Approaches to Extend Internet Addressing . . . . .	24

5. A System View on Address . . . . .	26
6. Issues in Extensions to Internet Addressing . . . . .	27
6.1. Limiting Address Semantics . . . . .	27
6.2. Complexity and Efficiency . . . . .	27
6.2.1. Repetitive encapsulation . . . . .	28
6.2.2. Compounding issues with header compression . . . . .	29
6.2.3. Introducing Path Stretch . . . . .	29
6.2.4. Complicating Traffic Engineering . . . . .	29
6.3. Security . . . . .	30
6.4. Fragility . . . . .	30
7. Summary of issues . . . . .	31
8. Conclusions . . . . .	33
9. Security Considerations . . . . .	34
10. IANA Considerations . . . . .	34
11. Informative References . . . . .	34
Acknowledgments . . . . .	44
Authors' Addresses . . . . .	44

## 1. Introduction

[I-D.jia-intarea-scenarios-problems-addressing] outlines scenarios and problems in Internet addressing through presenting a number of cases of communication that have emerged over the many years of utilizing the Internet and for which various extensions to the network interface-centric addressing of IPv6 have been developed. In order to continue the discussion on the emerging needs for addressing, initiated with [I-D.jia-intarea-scenarios-problems-addressing], this memo aims at identifying gaps between the Internet addressing model and desirable features that have been added by various extensions, in various contexts.

The approach to identifying the gaps is guided by key properties of Internet addressing, outlined in Section 2, namely (i) the fixed length of the IP addresses, (ii) the ambiguity of IP addresses semantic, while still (iii) providing limited IP address semantic support. Those properties are derived directly as a consequence of the respective standards that provide the basis for Internet addressing, most notably [RFC0791] for IPv4 and [RFC8200] for IPv6, respectively.

Those basic properties, and the potential issues that arise from those properties, give way to extensions that have been proposed over the course of deploying new Internet technologies. Section 3 discusses those extensions, summarized as gaps against the basic properties in Section 4.

Finally, this memo outlines issues that arise with the extension-driven approach to the basic Internet addressing, discussed in Section 6, arguing that any requirements for solutions that would revise the basic Internet addressing would require to address those issues.

## 2. Properties of Internet Addressing

As the Internet Protocol adoption has grown towards the global communication system we know today, its characteristics have evolved subtly, with [RFC6250] documenting various aspects of the IP service model and its frequent misconceptions, including Internet addressing. In this section, the three most acknowledged properties related to `_Internet addressing_` are detailed. Those are (i) fixed IP address length, (ii) ambiguous IP address semantic, and (iii) limited IP address semantic support.

Section 3 elaborates on various extensions that aim to expand Internet addressing beyond those properties; those extensions are positioned as intentions to close perceived gaps against those key properties.

### 2.1. Property 1: Fixed Address Length

The fixed IP address length is specified as a key property of the design of Internet addressing, with 32 bits for IPv4 ([RFC0791]), and 128 bits for IPv6 ([RFC8200]), respectively. Given the capability of the hardware at the time of IPv4 design, a fixed length address was considered as a more appropriate choice for efficient packet forwarding. Although the address length was once considered to be variable during the design of Internet Protocol Next Generation ("IPng", cf., [RFC1752]) in the 1990s, it finally inherited the design of IPv4 and adopted a fixed length address towards the current IPv6. As a consequence, the 128-bit fixed address length of IPv6 is regarded as a balance between fast forwarding (i.e., fixed length) and practically boundless cyberspace (i.e., enabled by using 128-bit addresses).

### 2.2. Property 2: Ambiguous Address Semantic

Initially, the meaning of an IP address has been to identify an interface on a network device, although, when [RFC0791] was written, there were no explicit definitions of the IP address semantic.

With the global expansion of the Internet protocol, the semantic of the IP address is commonly believed to contain at least two notions, i.e., the explicit 'locator', and the implicit 'identifier'. Because of the increasing use of IP addresses to both identify a node and to

indicate the physical or virtual location of the node, the intertwined address semantics of identifier and locator was then gradually observed and first documented in [RFC2101] as 'locator/identifier overload' property. With this, the IP address is used as an identification for host and server, very often directly used, e.g., for remote access or maintenance.

### 2.3. Property 3: Limited Address Semantic Support

Although IPv4 [RFC0791] did not add any semantic to IP addresses beyond interface identification (and location), time has proven that additional semantics are desirable (c.f., the history of 127/8 [HISTORY127] or the introduction of private addresses [RFC1918]), hence, IPv6 [RFC4291] introduced some form of additional semantics based on specific prefix values, for instance link-local addresses or a more structured multicast addressing. Nevertheless, systematic support for rich address semantics remains limited and basically prefix-based.

## 3. Filling Gaps through Extensions to Internet Addressing Properties

Over the years, a plethora of extensions has been proposed in order to move beyond the native properties of IP addresses, outlined in the previous section. The development of those extensions can be interpreted as filling gaps between the original properties of Internet addressing and desired new capabilities that those developing the extensions identified as being missing and yet needed and desirable.

### 3.1. Length Extensions

Extensions in this subsection aim at extending the property described in Section 2.1, i.e., the fixed IP address length.

When IPv6 was designed, the main objective was to create an address space that would not lead to the same situation as IPv4, namely to address exhaustion. To this end, while keeping the same addressing model like IPv4, IPv6 adopted a 128-bit address length with the aim of providing a sufficient and future-proof address space. The choice was also founded on the assumption that advances in hardware and Moore's law would still allow to make routing and forwarding faster, and the IPv6 routing table manageable.

We observe, however, that the rise of new use cases but also the number of new, e.g., industrial/home or small footprint devices, was possibly unforeseen. Sensor networks and more generally the Internet of Things (IoT) emerged after the core body of work on IPv6, thus different from IPv6 assumptions, 128-bit addresses were costly in

certain scenarios. On the other hand, given the huge investments that IPv6 deployment involved, certain solutions are expected to increase the addressing space of IPv4 in a compatible way, and thus extend the lifespan of the sunk investment on IPv4.

At the same time, it may also be possible to use variable and longer address lengths to address current networking demands. For example in content delivery networks, longer addresses such as URLs are required to fetch content, an approach that Information-Centric Networking (ICN) applied for any data packet sent in the network, using information-based addressing at the network layer. Furthermore, as an approach to address the routing challenges faced in the Internet, structured addresses may be used in order to avoid the need for routing protocols. Using variable length addresses allow as well to have shorter addresses. So for requirements for smaller network layer headers, shorter addresses could be used, maybe alleviating the need to compress other fields of the header. Furthermore, transport layer port numbers can be considered short addresses, where the high order bits of the extended address is the public IP of a NAT. Hence, in IoT deployments, the addresses of the devices can be really small and based on the port number, but they all share the global address of the gateway to make each one have a globally unique address.

### 3.1.1. Shorter Address Length

#### 3.1.1.1. Description:

In the context of IoT [RFC7228], where bandwidth and energy are very scarce resources, the static length of 128-bit for an IP address is more a hindrance than a benefit since 128-bit for an IP address may occupy a lot of space, even to the point of being the dominant part of a packet. In order to use bandwidth more efficiently and use less energy in end-to-end communication, solutions have been proposed that allow for very small network layer headers instead.

#### 3.1.1.2. Methodology:

- \* Header Compression/Translation: One of the main approaches to reduce header size in the IoT context is by compressing it. Such technique is based on a stateful approach, utilizing what is usually called a 'context' on the IoT sensor and the gateway for communications between an IoT device and a server placed somewhere in the Internet - from the edge to the cloud.

The role of the 'context' is to provide a way to 'compress' the original IP header into a smaller one, using shorter address information and/or dropping some field(s); the context here serves as a kind of dictionary.

- \* Separate device from locator identifier: Approaches that can offer customized address length that is adequate for use in such constrained domains are preferred. Using different namespaces for the 'device identifier' and the 'routing' or 'locator identifier' is one such approach.

#### 3.1.1.3. Examples

- \* Header Compression/Translation: Considering one base station is supposed to serve hundreds of user devices, maximizing the effectiveness for specific spectrum directly improves user quality of experience. To achieve the optimal utilization of the spectrum resource in the wireless area, the RObust Header Compression (ROHC) [RFC5795] mechanism, which has been widely adopted in cellular network like WCDMA, LTE, and 5G, utilizes header compression to shrink existing IPv6 headers onto shorter ones.

Similarly, header compression techniques for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) have been around for several years now, constituting a main example of using the notion of a 'shared context' in order to reduce the size of the network layer header ([RFC6282], [RFC7400], [ITU9959]). More recently, other compression solutions have been proposed for Low Power Wide Area Networks (LPWAN - [RFC8376]). Among them, the Static Context Header Compression (SCHC - [RFC8724]) generalized the compression mechanism developed by 6lo. Instead of a standard compression behavior implemented in all 6lo nodes, SCHC introduces the notion of rule shared by two nodes. The SCHC compression technique is generic and can be applied to IPv6 and above layers. Regarding the nature of the traffic, IPv6 addresses (source and destination) can be elided, partially sent, or replaced by a small index. Instead of the versatile IP packet, SCHC defines new packet formats dedicated to specific applications. SCHC rules are equivalence functions mapping this format to standard IP packets.

Also, constraints coming from either devices or carrier links would lead to mixed scenarios and compound requirements for extraordinary header compression. For native IPv6 communications on DECT ULE and MS/TP Networks [RFC6282], dedicated compression mechanisms are specified in [RFC8105] and [RFC8163], while the transmission of IPv6 packets over NFC and PLC, specifications are being developed in [I-D.ietf-6lo-nfc] and [I-D.ietf-6lo-plc].

- \* Separate device from locator identifier: Solutions such as proposed in [EIBP] and [I-D.ietf-lisp-rfc6830bis] can utilize a separation of device from locator, where only the latter is used for routing between the different domains using the same technology, therefore enabling the use of shorter addresses in the (possibly constrained) local environment. Device IDs used within such domains are carried as part of the payload by EIBP and hence can be of shorter size suited to the domain, while, for instance, in LISP a flexible address encoding [RFC8060] allows shorter addresses to be supported in the LISP control plane [I-D.ietf-lisp-rfc6833bis].

### 3.1.2. Longer Address Length

#### 3.1.2.1. Description

Historically, obtaining adequate address space is considered as the primary and raw motivation to invent IPv6. Longer address (more than 32-bit of IPv4 address), which can accommodate almost inexhaustible devices, used to be considered as the surest direction in 1990s. Nevertheless, to protect the sunk cost of IPv4 deployment, certain efforts focus on IPv4 address space depletion question but engineer IPv4 address length in a more practical way. Such effort, i.e., NAT (Network Address Translation), unexpectedly and significantly slows IPv6 deployment because of its high cost-effectiveness in practice.

Another crucial need for longer address lengths comes from "semantic extensions" to IP addresses, where the extensions themselves do not fit within the length limitation of the IP address. Section 3.3 discusses extensions which extend address semantics that are not limited by the IP address length.

This sub-section focuses on address length extensions that aim at reducing the IPv4 addresses depletion, while Section 3.3, i.e., address semantic extensions, may still refer to extensions when longer address length are suitable to accommodate different address semantic. See Section 3.3 for details of semantic-driven address lengthening.

#### 3.1.2.2. Methodology

- \* Split address zone by network realm: This methodology first split the network realm into two types: one public realm (i.e., the Internet), and innumerable private realms (i.e., local networks, which may be embedded and/or having different scope). Then, it splits the IP address space into two type of zones: global address zone (i.e., public address) and local address zone (e.g., private address, reserved address). Based on this, it is assumed that in

public realm, all devices attached to it should be assigned an address that belongs to the global address zone. While for devices attached to private realms, only addresses belonging to the local address zone will be assigned. Local realms may have different scope or even be embedded one in another, like for instance, light switches local network being part of the building local network, which in turn connects to the Internet. In the local realms address may have a pure identification purpose. For instance in last example, addresses of the light switches identify the switches themselves, while the building local network is used to locate them.

Given that the local address zone is not globally unique, certain mechanisms are designed to express the relationship between the global address zone (in public realm) and the local address zone (in any private realm). In this case, global addresses are used for forwarding when a packet is in the public realm, and local addresses are used for forwarding when a packet is in a private realms.

#### 3.1.2.3. Examples

- \* Split address zone by network realm: Network Address Translation (NAT), which was first laid out in [RFC2663], using private address and a stateful address binding to translate between the realms. As outlined in [RFC2663], basic address translation is usually extended to include port number information in the translation process, supporting bidirectional or simple outbound traffic only. Because the 16-bits port number is used in the address translation, NAT theoretically increase IPv4 address length from 32-bit to 48-bit, i.e., 281 trillion address space.

Similarly, EzIP [EzIP] expects to utilize a reserved address block, i.e., 240/4, and an IPv4 header option to include it. Based on this, it can be regarded as EzIP is carrying a hierarchical address with two parts, where each part is a partial 32-bit IPv4 address. The first part is a public address residing in the "address field" of the header from globally routable IPv4 pool [IPv4pool], i.e., ca. 3.84 billion address space. The second part is the reserved address residing in "option field" and belongs to the 240/4 prefix, i.e., ca.  $2^{28}=268$  million. Based on that, each EzIP deployment is tethered on the existing Internet via one single IPv4 address, and EzIP then have  $3.84B * 268M$  address, ca. 1,000,000 trillion. Collectively, the 240/4 can also be used as end point identifier and form an overlay network providing services parallel to the current Internet, yet independent of the latter in other aspects.



Compared to NAT, EzIP is able to establish a communication session from either side of it, hence being completely transparent, and facilitating a full end-to-end networking configuration.

### 3.1.3. Summary

Table 1 summarizes methodologies and examples towards filling gaps on IP address length extensions.

	Methodology	Examples
Shorter Address Length	Header compression/ translation	6LoWPAN, ROHC, SCHC
	Separate device from locator identifier	EIBP, LISP, ILNP, HIP
Longer Address Length	Split address zone by network realm	NAT, EzIP

Table 1: Summary Length Extensions

## 3.2. Identity Extensions

Extensions in this subsection attempt extending the property described in Section 2.2, i.e., 'locator/identifier overload' of the ambiguous address semantic.

From the perspective of Internet users, on the one hand, the implicit identifier semantic results in a privacy issue due to network behavior tracking and association. Despite that IP address assignments may be dynamic, they are nowadays considered as 'personal data' and as such undergoes privacy protection regulations like General Data Protection Regulation ("GDPR" [GDPR]). Hence, additional mechanisms are necessary in order to protect end user privacy.

For network regulation of sensitive information, on the other hand, dynamically allocated IP addresses are not sufficient to guarantee device or user identification. As such, different address allocation systems, with stronger identification properties are necessary where security and authentication are at highest priority. Hence, in order to protect information security within a network, additional mechanism are necessary to identify the users or the devices attached to the network.

### 3.2.1. Anonymous Address Identity

#### 3.2.1.1. Description

As discussed in Section 2.2, IP addresses reveal both 'network locations' as well as implicit 'identifier' information to both traversed network elements and destination nodes alike. This enables recording, correlation, and profiling of user behaviors and historical network traces, possibly down to individual real user identity. The IETF, e.g., in [RFC7258], has taken a clear stand on preventing any such pervasive monitoring means by classifying them as an attack on end users' right to be left alone (i.e., privacy). Regulations such as the EU's General Data Protection Regulation (GDPR) classifies, for instance, the 'online identifier' as personal data which must be carefully protected; this includes end users' IP addresses [GDPR].

Even before pervasive monitoring [RFC7258], IP addresses have been seen as something that some organizational owners of networked system may not want to reveal at the individual level towards any non-member of the same organization. Beyond that, if forwarding is based on semantic extensions, like other fields of the header, extension headers, or any other possible extension, if not adequately protected it may introduce privacy leakage and/or new attack vectors.

#### 3.2.1.2. Methodology:

- \* **Traffic Proxy:** Detouring the traffic to a trusted proxy is a heuristic solution. Since nodes between trusted proxy and destination (including the destination per se) can only observe the source address of the proxy, the 'identification' of the origin source can thereby be hidden. To obfuscate the nodes between origin and the proxy, the traffic on such route would be encrypted via a key negotiated either in-band or off-band. Considering that all applications' traffic in such route can be seen as a unique flow directed to the same 'unknown' node, i.e., the trusted proxy, eavesdroppers in such route have to make more efforts to correlate user behavior through statistical analysis even if they are capable of identifying the users via their source addresses. The protection lays in the inability to isolate single application specific flows. According to the methodology, such approach is IP version independent and works for both IPv4 and IPv6.
- \* **Source Address Rollover:** Privacy issues related to address 'identifier' semantic can be mitigated through regular change (beyond the typical 24 hours lease of DHCP). Due to the semantics of 'identifier' that an IP address carries, such approach promotes

to change the source IP address at a certain frequency. Under such methodology, the refresh cycling window may reach to a balance between privacy protection and address update cost. Due to the limited space that IPv4 contains, such approach usually works for IPv6 only.

- \* Private Address Spaces: Their introduction in [RFC1918] foresaw private addresses (assigned to specific address spaces by the IANA) as a means to communicate purely locally, e.g., within an enterprise, by separating private from public IP addresses. Considering that private addresses are never directly reachable from the Internet, hosts adopting private addresses are invisible and thus 'anonymous' for the Internet. Besides, hosts for purely local communication used the latter while hosts requiring public Internet service access would still use public IP addresses.
- \* Address Translation: The aforementioned original intention for using private IP addresses, namely for purely local communication, resulted in a lack of flexibility in changing from local to public Internet access on the basis of what application would require which type of service.

If eventually every end-system in an organization would require some form of public Internet access in addition to local one, an adequate number of public Internet addresses would be required for providing to all end systems. Instead, address translation enables to utilize many private IP addresses within an organization, while only relying on one (or few) public IP addresses for the overall organization.

In principle, address translation can be applied recursively. This can be seen in modern broadband access where Internet providers may rely on carrier-grade address translation for all their broadband customers, who in turn employ address translation of their internal home or office addresses to those (private again) IP addresses assigned to them by their network provider.

Two benefits arise from the use of (private to public IP) address translation, namely (i) the hiding of local end systems at the level of the (address) assigned organization, and (ii) the reduction of public IP addresses necessary for communication across the Internet. While the latter has been seen for long as a driver for address translation, we focus on the first issue in this section, also since we see such privacy benefit as well as objective as still being valid in addressing systems like IPv6 where address scarcity is all but gone [GNATCATCHER].

- \* Separate device from locator identifier: Solutions that make a clear separation between the routing locator and the identifier, can allow for a device ID of any size, which in turn can be encrypted by a network element deployed at the border of routing domain (e.g., access/edge router). Both source and end-domain addresses can be encrypted and transported, as in the routing domain, only the routing locator is used.

#### 3.2.1.3. Examples:

- \* Traffic Proxy: Although not initially designed as a traffic proxy approach, a Virtual Private Network (VPN [VPN]) is widely utilized for packets origin hiding as a traffic detouring methodology. As it evolved, VPN derivatives like WireGuard [WireGuard] have become a mainstream instance for user privacy and security enhancement.

With such methodology in mind, onion routing [ONION], instantiated in the TOR Project [TOR], achieves high anonymity through traffic hand over via intermediates, before reaching the destination. Since the architecture of TOR requires at least three proxies, none of them is aware of the entire route. Given that the proxies themselves can be deployed all over cyberspace, trust is not the prerequisite if proxies are randomly selected.

In addition, dedicated protocols are also expected to be customized for privacy improvement via traffic proxy. For example, Oblivious DNS over HTTPS (ODOH [ODOH]) use a third-party proxy to obscure identifications of user source addresses during DNS over HTTPS (DoH [RFC8484]) resolution. Similarly, Oblivious HTTP [OHTTP] involve proxy alike in the HTTP environment.

- \* Source Address Rollover: As for source address rollover, it has been standardized that IP addresses for Internet users should be dynamic and temporary every time they are being generated [RFC8981]. This benefits from the available address space in the case of IPv6, through which address generation or assignment should be unpredictable and stochastic for outside observers.

More radically, [EPHEMERALv6] advocates an 'ephemeral address', changing over time, for each process. Through this, correlating user behaviors conducted by different identifiers (i.e., source address) becomes much harder, if not impossible, if based on the IP packet header alone.

- \* Private Addresses: The use and assignment of private addresses for IPv4 is laid out in [RFC1918], while unique local addresses (ULAs) in IPv6 [RFC4193] take over the role of private address spaces in IPv4.

- \* Network Address Translation: Given address translation can be performed several times in cascade, NATs may exist as part of existing customer premise equipment (CPE), such as a cable or an Ethernet router, with private wired/wireless connectivity, or may be provided in a carrier environment to further translate ISP-internal private addresses to a pool of (assigned) public IP addresses. The latter is often dynamically assigned to CPEs during its bootstrapping.
- \* Separate device from locator identifier: EIBP [EIBP] utilizes a structured approach to addressing. It separates the routing ID from the device ID, where only the former is used for routing. As such, the device IDs can be encrypted, protecting the end device identity. Similarly, LISP uses separate namespaces for routing and identification allowing to 'hide' identifiers in encrypted LISP packets that expose only known routing information [RFC8061].

### 3.2.2. Authenticated Address Identity

#### 3.2.2.1. Description

In some scenarios (e.g., corporate networks) it is desirable to being able authenticate IP addresses in order to prevent malicious attackers spoofing IP addresses. This is usually achieved by using a mechanism that allows to prove ownership of the IP address.

#### 3.2.2.2. Methodology

- \* Self-certified addresses: This method is usually based on the use of nodes' public/private keys. A node creates its own interface ID (IID) by using a cryptographic hash of its public key (with some additional parameters). Messages are then signed using the nodes' private key. The destination of the message will verify the signature through the information in the IP address. Self-certification has the advantage that no third party or additional security infrastructure is needed. Any node can generate its own address locally and then only the address and the public key are needed to verify the binding between the public key and the address.

- \* Third party granted addresses: DHCP (Dynamic Host Configuration Protocol) is widely used to provide IP addresses, however, in its basic form, it does not perform any check and even an unauthorized user without the right to use the network can obtain an IP address. To solve this problem, a trusted third party has to grant access to the network before generating an address (via DHCP or other) that identifies the user. User authentication done securely either based on physical parameters like MAC addresses or based on an explicit login/password mechanism.

#### 3.2.2.3. Examples

- \* Self-certified Addresses: As an example of this methodology serves [RFC3972], defining IPv6 cryptographically Generated Addresses (CGA). A Cryptographically Generated Address is formed by replacing the least-significant 64 bits of an IPv6 address with the cryptographic hash of the public key of the address owner. Packets are then signed with the private key of the sender. Packets can be authenticate by the receiver by using the public key of the sender and the address of the sender. The original specifications have been already amended (cf., [RFC4581] and [RFC4982]) in order to support multiple (stronger) cryptographic algorithms.
- \* Third party granted addresses: [RFC3118] defines a DHCP option through which authorization tickets can be generated and newly attached hosts with proper authorization can be automatically configured from an authenticated DHCP server. Solutions exist where separate servers are used for user authentication like [UA-DHCP] and [RFC4014]. The former proposing to enhance the DHCP system using registered user login and password before actually providing an IP address lease and recording the MAC address of the device the user used to sign-in. The latter, couples the RADIUS authentication protocol ([RFC2865]) with DHCP, basically piggybacking RADIUS attributes in a DHCP sub-option, with the DHCP server contacting the RADIUS server to authenticate the user.

#### 3.2.3. Summary

Table 2, summarize the methodologies and the examples towards filling the gaps on identity extensions.

	Methodology	Examples
Anonymous Address Identity	Traffic Proxy	VPN, TOR, ODoH
	Source Address Rollover	SLAAC
	Private Address Spaces	ULA
	Address Translation	NAT
	Separate device from locator identifier	EIBP, LISP
Authenticated Address Identity	Self-certified Addresses	CGA
	Third party granted addresses	DHCP-Option

Table 2: Summary Identity Extensions

### 3.3. Semantic Extensions

Extensions in this subsection try extending the property described in Section 2.3, i.e., limited address semantic support.

As explained in Section 2.2, IP addresses carry both locator and identification semantic. Some efforts exist that try to separate these semantics either in different address spaces or through different address formats. Beyond just identification, location, and the fixed address size, other efforts extended the semantic through existing or additional header fields (or header options) outside the Internet address.

How much unique and globally routable an address should be? With the effect of centralization, edges communicate with (rather) local DCs, hence a unique address globally routable is not a requirement anymore. There is no need to use globally unique addresses all the time for communication, however, there is the need of having a unique address as a general way to communicate to any connected entity without caring what transmission networks the packets traverse.

### 3.3.1. Utilizing Extended Address Semantics

#### 3.3.1.1. Description

Several extensions have been developed to extend beyond the limited IPv6 semantics. Those approaches may include to apply structure to the address, utilize specific prefixes, or entirely utilize the IPv6 address for different semantics, while re-encapsulating the original packet to restore the semantics in another part of the network. For instance, structured addresses have the capability to introduce delimiters to identify semantic information in the header, therefore not constraining any semantic by size limitations of the address fields.

We note here that extensions often start out as being proposed as an extended header semantic, while standardization may drive the solution to adopt an approach to accommodate their semantic within the limitations of an IP address. This section does include examples of this kind.

#### 3.3.1.2. Methodology

\*Semantic prefixes: Semantic prefixes are used to separate the IPv6 address space. Through this, new address families, such as for information-centric networking [HICN], service routing or other semantically rich addressing, can be defined, albeit limited by the prefix length and structure as well as the overall length limitation of the IPv6 address.

\* Separate device/resource from locator identifier: The option to use separate namespaces for the device address would offer more freedom for the use of different semantics. For instance, the static binding of IP addresses to servers creates a strong binding between IP addresses and service/resources, which may be a limitation for large Content Distribution networks (CDNs) [FAYED21].

As an extreme form of separating resource from locator identifier, recent engineering approaches, described in [CLOUDFLARE\_SIGCOMM], decouple web service (semantics) from the routing address assignments by using virtual hosting capabilities, thereby effectively mapping possibly millions of services onto a single IP address.

\* Structured addressing: One approach to address the routing challenges faced in the Internet is the use of structured addresses, e.g., to void the need for routing protocols. Benefits of this approach can be significant, with the structured addresses



capturing the relative physical or virtual position of routers in the network as well as being variable in length. Key to the approach, however, is that the structured addresses capturing the relative physical or virtual position of routers in the network, or networks in an internetwork may not fit within the fixed and limited IP address length (cf., Section 3.1.2). Other structured approaches may be the use of application-specific structured binary components for identification, generalizing URL schema used for HTTP-level communication but utilized at the network level for traffic steering decisions.

- \* Localized forwarding semantics: Layer 2 hardware, such as SDN switches, are limited to the use of specific header fields for forwarding decisions. Hence, devising new localized forwarding mechanisms may be based on re-using differently existing header fields, such as the IPv6 source/destination fields, to achieve the desired forwarding behavior, while encapsulating the original packets in order to be restored at the local forwarding network boundary. Networks in those solutions are limited by the size of the utilized address field, e.g., 256 bits for IPv6, thereby limiting the way such techniques could be used.

#### 3.3.1.3. Examples

- \* Semantic prefixes: Newer approaches to IP anycast suggest the use of service identification in combination with a binding IP address model [SFCANYCAST] as a way to allow for metric-based traffic steering decisions; approaches for Service Function Chaining (SFC) [RFC7665] utilize the Network Service Header (NSH) information and packet classification to determine the destination of the next service.

Another example of the usage of different packet header extensions based on IP addressing is Segment Routing. In this case, the source chooses a path and encodes it in the packet header as an ordered list of segments. Segments are encoded using new Routing Extensions Header type, the Segment Routing Header (SRH), which contains the Segment List, similar to what is already specified in [RFC8200], i.e., a list of segment ID (SID) that dictate the path to follow in the network. Such segment IDs are coded as 128 bit IPv6 addresses [RFC8986].

Approaches such as [HICN] utilize semantic prefixing to allow for ICN forwarding behavior within an IPv6 network. In this case, an HICN name is the hierarchical concatenation of a name prefix and a name suffix, in which the name prefix is encoded as an IPv6 128 bits word and carried in IPv6 header fields, while the name suffix is encoded in transport headers fields such as TCP. However, it

is a challenge to determine which IPv6 prefixes should be used as name prefixes. In order to know which IPv6 packets should be interpreted based on an ICN semantic, it is desirable to be able to recognize that an IPv6 prefix is a name prefix, e.g. to define a specific address family (AF\_HICN, b0001::/16). This establishment of a specific address family allows the management and control plane to locally configure HICN prefixes and announce them to neighbors for interconnection.

- \* Separate device from locator identifier: EIBP [EIBP] separates the routing locator from the device identifier, relaxing therefore any semantic constraints on the device identifier. Similarly, LISP uses a flexible encoding named LISP Canonical Address Format (LCAF [RFC8061]), which allows to associate to routing locators any possible form (and length) of identifier. ILNP [RFC6740] introduces as well a different semantic of IP addresses, while aligning to the IPv6 address format (128 bits). Basically, ILNP introduces a sharper logical separation between the 64 most significant bits and the 64 least significant bits of an IPv6 address. The former being a global locator, while the latter being an identifier that can have different semantics (rather than just being an interface identifier).
- \* Structured addressing: Network topology captures the physical connectivity among devices in the network. There is a structure associated with the topology. Examples are the core-distribution-access router structure commonly used in enterprise networks and clos topologies that are used to provide multiple connections between Top of Rack (ToR) devices and multiple layers of spine devices. Internet service providers use a tier structure that defines their business relationships. A clear structure of connected networks can be noticed in the Internet. EIBP [EIBP] proposes to leverage the physical structure (or a virtual structure overlaid on the physical structure) to auto assign addresses to routers in a network or networks in an internetwork to capture their relative position in the physical/virtual topology. EIBP proposes to administratively identify routers/networks with a tier value based on the structure.
- \* Localized forwarding semantics: Approaches such as those outlined in [REED] suggest using a novel forwarding semantic based on path information carried in the packet itself, said path information consists in a fixed size bit-field (see [REED] for more information on how to represent the path information in said bit-field). In order to utilize existing, e.g., SDN-based, forwarding switches, the direct use of the IPv6 source/destination address is suggested for building appropriate match-action rules (over the suitable binary information representing the local output ports),

while preserving the original IPv6 information in the encapsulated packet. As mentioned above, such use of the existing IPv6 address fields limits the size of the network to a maximum of 256 bits (therefore paths in the network over which such packets can be forwarded). [ICNIP], however, goes a step further by suggesting to use the local forwarding as direct network layer mechanism, removing the IP packet and only leaving the transport/application layer, with the path identifier constituting the network-level identifier albeit limited by using the existing IP header for backward compatibility reasons (the next section outlines the removal of this limitation).

### 3.3.2. Utilizing Existing or Extended Header Semantics

#### 3.3.2.1. Description:

While the former sub-section explored extended address semantic, thereby limiting any such extended semantic with that of the existing IPv6 semantic and length, additional semantics may also be placed into the header of the packet or the packet itself, utilized for the forwarding decision to the appropriate endpoint according to the extended semantic.

Reasons for embedding such new semantics may be related to traffic engineering since it has long been shown that the IP address itself is not enough to steer traffic properly since the IP address itself is not semantically rich enough to adequately describe the forwarding decision to be taken in the network, not only impacting WHERE the packet will need to go but also HOW it will need to be sent.

#### 3.3.2.2. Methodology:

- \* In-Header extensions: One way to add additional semantics besides the address fields is to use other fields already present in the header.
- \* Headers option extensions: Another mechanism to add additional semantics is to actually add additional fields, e.g., through Header Options in IPv4 or through Extension Headers in IPv6.
- \* Re-encapsulation extension: A more radical approach for additional semantics is the use of a completely new header that is designed so to carry the desired semantics in an efficient manner (often as a shim header).
- \* Structured addressing: Similar to the methodology that structures addresses within the limitations of the IPv6 address length, outlined in the previous sub-sections, structured addressing can

also be applied within existing or extended header semantics, e.g., utilizing a dedicated (extension) header to carry the structured address information.

- \* Localized forwarding semantics: This set of solutions applies capabilities of newer (programmable) forwarding technology, such as [P4], to utilize any header information for a localized forwarding decision. This removes any limitation to use existing header or address information for embedding a new address semantic into the transferred packet.

#### 3.3.2.3. Examples:

- \* In-Header extensions: In order to allow additional semantic with respect to the pure Internet addressing, the original design of IPv4 included the field 'Type of Service' [RFC2474], while IPv6 introduced the 'Flow label' and the 'Traffic Class' [RFC8200]. In a certain way, those fields can be considered 'semantic extensions' of IP addresses, and they are 'in-header' because natively present in the IP header (differently from options and extension headers). However, they proved not to be sufficient. Very often a variety of network operation are performed on the well-known 5-tuple (source and destination addresses; source and destination port number; and protocol number). In some contexts all of the above mentioned fields are used in order to have a very fine grained solution ([RFC8939]).
- \* Headers option extensions: Header options have been largely under-exploited in IPv4. However, the introduction of the more efficient extension header model in IPv6 along with technology progress made the use of header extensions more widespread in IPv6. Segment Routing re-introduced the possibility to add path semantic to the packet by encoding a loosely defined source routing ([RFC8402]). Similarly, in the aim to overcome the inherent shortcoming of the multi-homing in the IP context, SHIM6 ([RFC5533]) also proposed the use of an extension header able to carry multi-homing information which cannot be accommodated natively in the IPv6 header.

To serve a moving endpoint, mechanisms like Mobile IPv6 [RFC6275] are used for maintaining connection continuity by a dedicated IPv6 extension header. In such case, the IP address of the home agent in Mobile IPv6 is basically an identification of the on-going communication. In order to go beyond the interface identification model of IP, the Host Identity Protocol (HIP) tries to introduce an identification layer to provide (as the name says) host identification. The architecture here relies on the use of another type of extension header [RFC7401].

- \* Re-encapsulation extension: Differently from the previous approach, re-encapsulation prepends complete new IP headers to the original packet introducing a completely custom shim header between the outer and inner header. This is the case for LISP, adding a LISP specific header right after an IP+UDP header ([I-D.ietf-lisp-rfc6830bis]). A similar design is used by VxLAN ([RFC7348]) and GENEVE ([RFC8926]), even if they are designed for a data center context. IP packets can also be wrapped with headers using more generic and semantically rich names, for instance with ICN [ICNIP].
- \* Structured addressing: Solutions such as those described in the previous sub-section, e.g., EIBP [EIBP], can provide structured addresses that are not limited to the IPv6 address length but instead carry the information in an extension header to remove such limitation.

Also Information-Centric Networking (ICN) naming approaches usually introduce structures in the (information) names without limiting themselves to the IP address length; more so, ICN proposes its own header format and therefore radically breaks with not only IP addressing semantic but the format of the packet header overall. For this, approaches such as those described in [RFC8609] define a TLV-based binary application component structure that is carried as a 'name' part of the CCN messages. Such a name is a hierarchical structure for identifying and locating a data object, which contains a sequence of name components. Names are coded based on 2-level nested Type-Length-Value (TLV) encodings, where the name-type field in the outer TLV indicates this is a name, while the inner TLVs are name components including a generic name component, an implicit SHA-256 digest component and a SHA-256 digest of Interest parameters. For textual representation, URIs are normally used to represent names, as defined in [RFC3986].

In geographic addressing, position based routing protocols use the geographic location of nodes as their addresses, and packets are forwarded when possible in a greedy manner towards the destination. For this purpose, the packet header includes a field coding the geographic coordinates (x, y, z) of the destination node, as defined in [RFC2009]. Some proposals also rely on extra fields in the packet header to code the distance towards the destination, in which case only the geographic coordinates of neighbors are exchanged. This way the location of the destination is protected even if routing packets are eavesdropped.

- \* Localized forwarding semantics: Unlike the original suggestion in [REED] to use existing SDN switches, the proliferation of P4 [P4] opens up the possibility to utilize a locally limited address semantic, e.g., expressed through the path identifier, as an entirely new header (including its new address) with an encapsulation of the IP packet for E2E delivery (including further delivery outside the localized forwarding network or positioning the limited address semantic directly as the network address semantic for the packet, i.e., removing any IP packet encapsulation from the forwarded packet, as done in [ICNIP]). Removing the IPv6 address size limitation by not utilizing the existing IP header for the forwarding decision also allows for extensible length approaches for building the path identifier with the potential for increasing the supported network size. On the downside, this approach requires to encapsulate the original IP packet header for communication beyond the local domain in which the new header is being used, such as discussed in the previous point above on 're-encapsulation extension'.

### 3.3.3. Summary

Table 3, summarize the methodologies and the examples towards filling the gaps on semantic extensions.

	Methodology	Examples
Utilizing Extended Address Semantics	Semantic prefixes	HICN
	Separate device from locator identifier	EIBP, ILNP, LISP, HIP
	Structured addressing	EIBP, ILNP
	Localized forwarding semantics	REED
Utilizing Existing or Extended Header Semantics	In-Header extensions	DetNet
	Headers option extensions	SHIM6, SRv6, HIP
	Re-encapsulation extension	VxLAN, ICNIP
	Structured addressing	EIBP
	Localized forwarding semantics	REED

Table 3: Summary Semantic Extensions

#### 4. Overview of Approaches to Extend Internet Addressing

The following Table 4 describes the objectives of the extensions discussed in this memo with respect to the properties of Internet addressing (Section 2). As summarized, extensions may aim to extend one property of the Internet addressing, or extend other properties at the same time.

	Length Extension	Identity Extension	Semantic Extension
6LoWPAN	x		
ROHC	x		

EzIP	x		
TOR		x	
ODoH		x	
SLAAC		x	
CGA		x	x
NAT	x	x	
HICN		x	x
ICNIP	x	x	x
CCNx names	x	x	x
EIBP	x	x	x
Geo addressing	x		x
REED	x (with P4)		x
DetNet		x	
Mobile IP			x
SHIM6			x
SRv6			x
HIP		x	x
VxLAN		x	x
LISP		x	x
SFC		x	x

Table 4: Relationship between Extensions and Internet Addressing



## 5. A System View on Address

In the following, we investigate in which parts of the overall Internet system extensions have been proposed and developed. For this, we divide the possible innovation across two dimensions:

- \* Horizontal: Internet edge vs core. The criticality, scale, investment on the core of the Internet makes it more difficult to introduce innovation, while at the edges there is more flexibility. As general purpose processors have drastically improved in performance, data-plane features can be implemented in software. At the edge of the Internet, it is easier to introduce innovation for several reasons: Economics, faster ROI because of faster deployment; No need of large scale deployment (and hence less standardization effort); less stakeholders involved (sometimes just one, see following point). Furthermore, the fact that the edge is a place where there is less coordination and cooperation from the core, is another factor that eases the innovation.
- \* Vertical: at which layer of the protocol stack. The difficulty to innovate varies as well depending at which layer the innovation takes place. One thing is to innovate at application layer where the app developer has large degree of freedom, another is to innovate at network layer, which is more constrained because of its central point in the architecture. Innovation at higher layer sometimes leads to walled gardens (aka limited domains [RFC8799]). Indeed because of the centralization phenomena, an actor offering a certain service may very well develop and deploy a custom technology that does not need to be actually standardized because it is done for its own internal usage.
- \* Horizontal vs Vertical Innovation:
  - In the public Internet, core innovation at lower layer is harder, often reduced to app-level innovation or building an overlay limited domain (aka a walled garden).
  - At the edges it is easier to innovate at lower layers (more vertical flexibility) but some form of adaptation is needed if global reachability is wanted.

Despite these two orthogonal dimensions, innovation does not happen either horizontally or vertically, rather in both dimensions simultaneously at various degree.

## 6. Issues in Extensions to Internet Addressing

While the extensions to the original Internet properties, discussed in Section 3, demonstrate the benefits of more flexibility in addressing, they also bring with them a number of issues, which are discussed in the following section. To this end, the problems hereafter outlined link to the approaches to extensions summarized in Section 4. These issues may not be present all the time and everywhere, since as explained in Section 5, extensions are developed and deployed in different part of the Internet, which may worsen things.

### 6.1. Limiting Address Semantics

Many approaches changing the semantics of communication, e.g., through separating host identification from network node identification [RFC7401], separating the device identifier from the routing locator ([EIBP], [I-D.ietf-lisp-introduction]), or through identifying content and services directly [HICN], are limited by the existing packet size and semantic constraints of IPv6, e.g., in the form of its source and destination network addresses.

While approaches such as [ICNIP] may override the addressing semantics, e.g., by replacing IPv6 source and destination information with path identification, a possible unawareness of endpoints still requires the carrying of other address information as part of the payload.

Also, the expressible service or content semantic may be limited, as in [HICN] or the size of supported networks [REED] due to relying on the limited bit positions usable in IPv6 addresses.

### 6.2. Complexity and Efficiency

A crucial issue is the additional complexity introduced for realizing the additional addressing semantics. This is particularly an issue since we see those additional semantics particularly at the edge of the Internet, utilizing the existing addressing semantic of the Internet to interconnect the domains that require those additional semantics.

Furthermore, any additional complexity often comes with an efficiency and cost penalty, particularly at the edge of the network, where resource constraints may play a significant role. Compression processes, taking [ROHC] as an example, require additional resources both for the sender generating the compressed header but also the gateway linking to the general Internet by re-establishing the full IP header.

Conversely, the performance requirements of core networks, in terms of packet processing speed, makes the accommodation of extensions to addressing often prohibitive. This is not only due to the necessary extra processing that is specific to the extension, but also due to the complexity that will need to be managed in doing so at significantly higher speeds than at the edge of the network. The observations on the dropping of packets with IPv6 extension headers in the real world is (partially) due to such a implementation complexity [RFC7872].

Another example for lowering the efficiency of packet forwarding is the routing in systems like TOR [TOR]. As detailed before, traffic in TOR, for anonymity purposes, should be handed over by at least three intermediates before reaching the destination. Frequent relaying enhances the privacy, however, because such kind of solutions are implemented at application level, they come at the cost of lower communication efficiency. May be a different privacy enhanced address semantic would enable efficient implementation of TOR-like solutions at network layer.

#### 6.2.1. Repetitive encapsulation

Repetitive encapsulation is an issue since it bloats the packets size due to additional encapsulation headers. Addressing proposals such as those in [ICNIP] utilize path identification within an alternative forwarding architecture that acts upon the provided path identification. However, due to the limitation of existing flow-based architectures with respect to the supported header structures (in the form of IPv4 or IPv6 headers), the new routing semantics are being inserted into the existing header structure, while repeating the original, sender-generated header structure, in the payload of the packet as it traverses the local domain, effectively doubling the per-packet header overhead.

The problem is also present in a number of solutions tackling different issues, e.g., mobility [I-D.ietf-lisp-mn], DC networking ([RFC8926], [RFC7348], [I-D.ietf-intarea-gue]), traffic engineering [RFC8986], and privacy ([TOR], [SPHINX]). Certainly these solutions are able to avoid other issues, like path lengthening or privacy issues, as described before, but they come at the price of multiple encapsulations that reduce the effective payload. This, not only hampers efficiency in terms of header-to-payload ratio, but also introduces 'encapsulation points', which in turn add complexity to the (often edge) network as well as fragility due to the addition of possible failure points; this aspect is discussed in further details in Section 6.4.

#### 6.2.2. Compounding issues with header compression

IP header overhead requires header compression in constrained environments, such as wireless sensor networks and IoT in general. Together with fragmentation, both tasks constitute significant energy consumption, as shown in [HEADER\_COMP\_ISSUES1], negatively impacting resource limited devices that often rely on battery for operation. Further, the reliance on the compression/decompression points creates a dependence on such gateways, which may be a problem for intermittent scenarios.

According to the implementation of `_contiki-ng_` [CONTIKI], an example of operating system for IoT devices, the source codes for 6LowPan requires at least 600Kb to include a header compression process. In certain use cases, such requirement can be an obstacle for extremely constrained devices, especially for the RAM and energy consumption.

#### 6.2.3. Introducing Path Stretch

Mobile IP [RFC6275], which was designed for connection continuity in the face of moving endpoints, is a typical case for path stretch. Since traffic must follow a triangular route before arriving at the destination, such detour routing inevitably impacts transmission efficiency as well as latency.

#### 6.2.4. Complicating Traffic Engineering

While many extensions to the original IP address semantic target to enrich the decisions that can be taken to steer traffic, according to requirements like QoS, mobility, chaining, compute/network metrics, flow treatment, path usage, etc., the realization of the mechanisms as individual solutions likely complicates the original goal of traffic engineering when individual solutions are being used in combination. Ultimately, this may even prevent the combined use of more than one mechanism and/or policy with a need to identify and prevent incompatibilities of mechanisms. Key here is not the issue arising from using conflicting traffic engineering policies, rather conflicting realizations of policies that may well generally work well alongside ([ROBUSTSDN], [TRANSACTIONSDN]).

This not only increases fragility, as discussed separately in Section 6.4, but also requires careful planning of which mechanisms to use and in which combination, likely needing human-in-the-loop approaches alongside possible automation approaches for the individual solutions.

### 6.3. Security

The properties described in Section 2 have, obviously, also consequences in terms of security and privacy related issues, as already mentioned in other parts of this document.

For instance, in the effort of being somehow backward compatible, HIP [RFC7401] uses a 128-bit Host Identity, which may be not sufficiently cryptographically strong in the future because of the limited size (future computational power may erode 128-bit security). Similarly, CGA [RFC3972] also aligns to the 128-bit limit, but may use only 59 bits of them, hence, the packet signature may not be sufficiently robust to attacks [I-D.rafiiee-6man-cga-attack].

IP addresses, even temporary ones meant to protect privacy, have been long recognized as a 'Personal Identification Information' that allows even to geolocate the communicating endpoints [RFC8280]. The use of temporary addresses provides sufficient privacy protection only if the renewal rate is high [EPHEMERALv6]. However, this causes additional issues, like the large overhead due to the Duplicate Address Detection, the impact on the Neighbor Discovery mechanism, in particular the cache, which can even lead to communication disruption. With such drawbacks, the extensions may even lead to defeat the target, actually lowering security rather than increasing it.

The introduction of alternative addressing semantics has also been used to help in (D)DoS attacks mitigation. This leverages on changing the service identification model so to avoid topological information exposure, making the potential disruptions likely remain limited [ADDRLESS]. However, this increased robustness to DDoS comes at the price of important communication setup latency and fragility, as discussed next.

### 6.4. Fragility

From the extensions discussed in Section 3, it is evident that having alternative or additional address semantic and formats available for making routing as well as forwarding decisions dependent on these, is common place in the Internet. This, however, adds many extension-specific translation/adaptation points, mapping the semantic and format in one context into what is meaningful in another context, but also, more importantly, creating a dependency towards an additional component, often without explicit exposure to the endpoints that originally intended to communicate.

For instance, the re-writing of IP addresses to facilitate the use of private address spaces throughout the public Internet, realized through network address translators (NATs), conflicts with the end-to-end nature of communication between two endpoints. Additional (flow) state is required at the NAT middle-box to smoothly allow communication, which in turn creates a dependency between the NAT and the end-to-end communication between those endpoints, thus increasing the fragility of the communication relation.

A similar situation arises when supporting constrained environments through a header compression mechanism, adding the need for, e.g., a ROHC [RFC5795] element in the communication path, with communication-related compression state being held outside the communicating endpoints. Failure will introduce some inefficiencies due to context regeneration, which may affect the communicating endpoints, increasing fragility of the system overall.

Such translation/adaptation between semantic extensions to the original 'semantic' of an IP address is generally not avoidable when accommodating more than a single universal semantic. However, the solution-specific nature of every single extension is likely to noticeably increase the fragility of the overall system, since individual extensions will need to interact with other extensions that may be deployed in parallel, but were not designed taking into account such deployment scenario (cf., [I-D.ietf-intarea-tunnels]). Considering that extensions to traditional per-hop-behavior (based on IP addresses) can essentially be realized over almost 'any' packet field, the possible number of conflicting behaviors or diverging interpretation of the semantic and/or content of such fields, among different extensions, may soon become an issue, requiring careful testing and delineation at the boundaries of the network within which the specific extension has been realized.

## 7. Summary of issues

Table 5, derived from Section 6, summarizes the issues related to each extension. While each extension involves at least one issue, some others, like ICNIP, may create several issues at the same time.

	Limiting Address Semantics	Complexity and Efficiency	Security	Fragility
6LoWPAN		x		x
ROHC		x		x

EzIP		x		
TOR		x		x
ODoH		x		
SLAAC		x		
CGA	x		x	
NAT		x		x
HICN	x			
ICNIP	x	x		
CCNx name	x			
EIBP				x
Geo addressing	x			x
REED	x			
DetNet		x		
Mobile IP		x		x
SHIM6				x
SRv6				x
HIP			x	x
VxLAN		x		
LISP		x		x
SFC		x		x

Table 5: Issues in Extensions to Internet Addressing

## 8. Conclusions

The examples of extensions discussed in Section 3 to the original Internet addressing scheme show that extensibility beyond the original model (and its underlying per-hop behavior) is a desired capability for networking technologies and has been so for a long time. Generally, we can observe that those extensions are driven by the requirements of stakeholders, expecting a desirable extended functionality from the introduction of the specific extension. If interoperability is required, those extensions require standardization of possibly new fields, new semantics as well as (network and/or end system) operations alike.

The issues we identified in this document with the extension-specific solution approach, point to the need for a discussion on Internet addressing, as formulated in the companion document [I-D.jia-intarea-scenarios-problems-addressing] that formalizes the problem statement through scenarios that highlight the shortcomings of the Internet addressing model.

It is our conclusion that the existence of the many extensions to the original Internet addressing is clear evidence for gaps that have been identified over time by the wider Internet community, each of which come with a raft of issues that we need to deal with daily: We believe that it is time to develop an architectural but more importantly a sustainable approach to make Internet addressing extensible in order to capture the many new use cases that will still be identified for the Internet to come.

To jumpstart any such effort from an addressing perspective, it will be key to suitably define what an address is at which layer of the overall system, let alone the network layer. We argue that any answer to this question must be derived from what features we may want from the network instead of being guided by the answers that the Internet can give us today, e.g., being a mere ephemeral token for accessing PoP-based services (as indicated in related arch-d mailing list discussions).

This is not to 'second guess' the market and its possible evolution, but to outline clear features from which to derive clear principles for a design. Any such design must not skew the technical capabilities of addressing to the current economic situation of the Internet since this bears the danger of locking down innovation capabilities as an outcome of those technical limitations introduced. Instead, addressing must be aligned with enabling the model of permissionless innovation that the IETF has been promoting, ultimately enabling the serendipity of new applications that has led to many of those applications we can see in the Internet today. Most



importantly, any inaction on our side in that regard will only compound the issues identified, eventually hampering the future Internet's readiness for those new uses.

## 9. Security Considerations

The present memo does not introduce any new technology and/or mechanism and as such does not introduce any security threat to the TCP/IP protocol suite.

As an additional note, and as discussed in this document, security and privacy aspects were not considered as part of the key properties for Internet addressing, which led to the introduction of a number of extensions intending to fix those gaps. The analysis presented in this memo (non-exhaustively) shows those issues are either solved in an ad-hoc manner at application level, or at transport layer, while at network level only few extensions tackling specific aspects exist, albeit often with limitations due to the adherence to the Internet addressing model and its properties.

## 10. IANA Considerations

This document does not include any IANA request.

## 11. Informative References

[ADDRLESS] Hao, S., Liu, R., Weng, Z., Chang, D., Bao, C., and X. Li, "Addressless: A new internet server model to prevent network scanning", PLOS ONE Vol. 16, pp. e0246293, DOI 10.1371/journal.pone.0246293, February 2021, <<https://doi.org/10.1371/journal.pone.0246293>>.

[CLOUDFLARE\_SIGCOMM] Fayed, M., Bauer, L., Giotsas, V., Kerola, S., Majkowski, M., Odintsov, P., Sitnicki, J., Chung, T., Levin, D., Mislove, A., Wood, C., and N. Sullivan, "The ties that unbind: decoupling IP from web services and sockets for robust addressing agility at CDN-scale", Proceedings of the 2021 ACM SIGCOMM 2021 Conference, DOI 10.1145/3452296.3472922, August 2021, <<https://doi.org/10.1145/3452296.3472922>>.

[CONTIKI] "Contiki-NG: The OS for Next Generation IoT Devices", n.d., <<https://github.com/contiki-ng/contiki-ng>>.

- [EIBP] Shenoy, S Chandraiah, P Willis, N., "A Structured Approach to Routing in the Internet", June 2021, <First Intl Workshop on Semantic Addressing and Routing for Future Networks>.
- [EPHEMERALv6] Gont, F. and G. Gont, "IPv6 Addressing Considerations", Work in Progress, Internet-Draft, draft-gont-v6ops-ipv6-addressing-considerations-01, 21 February 2021, <<https://www.ietf.org/archive/id/draft-gont-v6ops-ipv6-addressing-considerations-01.txt>>.
- [EzIP] Chen, A. Y., Ati, R. R., Karandikar, A., and D. R. Crowe, "Adaptive IPv4 Address Space", Work in Progress, Internet-Draft, draft-chen-ati-adaptive-ipv4-address-space-10, 8 December 2021, <<https://www.ietf.org/archive/id/draft-chen-ati-adaptive-ipv4-address-space-10.txt>>.
- [FAYED21] Fayed, M., Bauer, L., Giotsas, V., Kerola, S., Majkowski, M., Odintsov, P., Sitnicki, J., Chung, T., Levin, D., Mislove, A., Wood, C., and N. Sullivan, "The ties that unbind: decoupling IP from web services and sockets for robust addressing agility at CDN-scale", Proceedings of the 2021 ACM SIGCOMM 2021 Conference, DOI 10.1145/3452296.3472922, August 2021, <<https://doi.org/10.1145/3452296.3472922>>.
- [GDPR] Voigt, P. and A. von dem Bussche, "The EU General Data Protection Regulation (GDPR)", Springer International Publishing book, DOI 10.1007/978-3-319-57959-7, 2017, <<https://doi.org/10.1007/978-3-319-57959-7>>.
- [GNATCATCHER] "Global Network Address Translation Combined with Audited and Trusted CDN or HTTP-Proxy Eliminating Reidentification", n.d., <<https://github.com/bslassey/ip-blindness>>.
- [HEADER\_COMP\_ISSUES1] Mesrinejad, F., Hashim, F., Noordin, N., Rasid, M., and R. Abdullah, "The effect of fragmentation and header compression on IP-based sensor networks (6LoWPAN)", The 17th Asia Pacific Conference on Communications, DOI 10.1109/apcc.2011.6152926, October 2011, <<https://doi.org/10.1109/apcc.2011.6152926>>.

- [HICN] Muscariello, L., "Hybrid Information-Centric Networking: ICN inside the Internet Protocol", March 2018, <<https://datatracker.ietf.org/meeting/interim-2018-icnrg-01/materials/slides-interim-2018-icnrg-01-sessa-hybrid-icn-hicn-luca-muscariello>>.
- [HISTORY127] "History of 127/8 as localhost/loopback addresses", n.d., <<https://elists.isoc.org/pipermail/internet-history/2021-January/006920.html>>.
- [I-D.ietf-6lo-nfc] Choi, Y., Hong, Y., Youn, J., Kim, D., and J. Choi, "Transmission of IPv6 Packets over Near Field Communication", Work in Progress, Internet-Draft, draft-ietf-6lo-nfc-17, 23 August 2020, <<https://www.ietf.org/archive/id/draft-ietf-6lo-nfc-17.txt>>.
- [I-D.ietf-6lo-plc] Hou, J., Liu, B., Hong, Y., Tang, X., and C. E. Perkins, "Transmission of IPv6 Packets over PLC Networks", Work in Progress, Internet-Draft, draft-ietf-6lo-plc-10, 17 February 2022, <<https://www.ietf.org/archive/id/draft-ietf-6lo-plc-10.txt>>.
- [I-D.ietf-intarea-gue] Herbert, T., Yong, L., and O. Zia, "Generic UDP Encapsulation", Work in Progress, Internet-Draft, draft-ietf-intarea-gue-09, 26 October 2019, <<https://www.ietf.org/archive/id/draft-ietf-intarea-gue-09.txt>>.
- [I-D.ietf-intarea-tunnels] Touch, J. and M. Townsley, "IP Tunnels in the Internet Architecture", Work in Progress, Internet-Draft, draft-ietf-intarea-tunnels-10, 12 September 2019, <<https://www.ietf.org/archive/id/draft-ietf-intarea-tunnels-10.txt>>.
- [I-D.ietf-lisp-introduction] Cabellos, A. and D. S. (Ed.), "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", Work in Progress, Internet-Draft, draft-ietf-lisp-introduction-15, 20 September 2021, <<https://www.ietf.org/archive/id/draft-ietf-lisp-introduction-15.txt>>.

[I-D.ietf-lisp-mn]

Farinacci, D., Lewis, D., Meyer, D., and C. White, "LISP Mobile Node", Work in Progress, Internet-Draft, draft-ietf-lisp-mn-11, 30 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-lisp-mn-11.txt>>.

[I-D.ietf-lisp-rfc6830bis]

Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos, "The Locator/ID Separation Protocol (LISP)", Work in Progress, Internet-Draft, draft-ietf-lisp-rfc6830bis-36, 18 November 2020, <<https://www.ietf.org/archive/id/draft-ietf-lisp-rfc6830bis-36.txt>>.

[I-D.ietf-lisp-rfc6833bis]

Farinacci, D., Maino, F., Fuller, V., and A. Cabellos, "Locator/ID Separation Protocol (LISP) Control-Plane", Work in Progress, Internet-Draft, draft-ietf-lisp-rfc6833bis-30, 18 November 2020, <<https://www.ietf.org/archive/id/draft-ietf-lisp-rfc6833bis-30.txt>>.

[I-D.jia-intarea-scenarios-problems-addressing]

Jia, Y., Trossen, D., Iannone, L., Shenoy, N., Mendes, P., 3rd, D. E. E., and P. Liu, "Challenging Scenarios and Problems in Internet Addressing", Work in Progress, Internet-Draft, draft-jia-intarea-scenarios-problems-addressing-02, 23 October 2021, <<https://www.ietf.org/archive/id/draft-jia-intarea-scenarios-problems-addressing-02.txt>>.

[I-D.rafiiee-6man-cga-attack]

Rafiiee, H. and C. Meinel, "Possible Attack on Cryptographically Generated Addresses (CGA)", Work in Progress, Internet-Draft, draft-rafiiee-6man-cga-attack-03, 8 May 2015, <<https://www.ietf.org/archive/id/draft-rafiiee-6man-cga-attack-03.txt>>.

[ICNIP]

Trossen, D., Robitzsch, S., Reed, M., Al-Naday, M., and J. Riihijarvi, "Internet Services over ICN in 5G LAN Environments", Work in Progress, Internet-Draft, draft-trossen-icnrg-internet-icn-5glan-04, 1 October 2020, <<https://www.ietf.org/archive/id/draft-trossen-icnrg-internet-icn-5glan-04.txt>>.

- [IPv4pool] "IANA IPv4 Address Space Registry", n.d.,  
<<https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml>>.
- [ITU9959] Badenhop, C., Fuller, J., Hall, J., Ramsey, B., and M. Rice, "Evaluating ITU-T G.9959 Based Wireless Systems Used in Critical Infrastructure Assets", IFIP Advances in Information and Communication Technology pp. 209-227, DOI 10.1007/978-3-319-26567-4\_13, 2015,  
<[https://doi.org/10.1007/978-3-319-26567-4\\_13](https://doi.org/10.1007/978-3-319-26567-4_13)>.
- [ODoH] Kinnear, E., McManus, P., Pauly, T., Verma, T., and C. A. Wood, "Oblivious DNS Over HTTPS", Work in Progress, Internet-Draft, draft-pauly-dprive-oblivious-doh-11, 17 February 2022, <<https://www.ietf.org/archive/id/draft-pauly-dprive-oblivious-doh-11.txt>>.
- [OHTTP] Thomson, M. and C. A. Wood, "Oblivious HTTP", Work in Progress, Internet-Draft, draft-thomson-http-oblivious-02, 24 August 2021, <<https://www.ietf.org/archive/id/draft-thomson-http-oblivious-02.txt>>.
- [ONION] Goldschlag, D., Reed, M., and P. Syverson, "Onion routing", Communications of the ACM Vol. 42, pp. 39-41, DOI 10.1145/293411.293443, February 1999,  
<<https://doi.org/10.1145/293411.293443>>.
- [P4] Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., and D. Walker, "P4: programming protocol-independent packet processors", ACM SIGCOMM Computer Communication Review Vol. 44, pp. 87-95, DOI 10.1145/2656877.2656890, July 2014,  
<<https://doi.org/10.1145/2656877.2656890>>.
- [REED] Reed, M., Al-Naday, M., Thomos, N., Trossen, D., Petropoulos, G., and S. Spirou, "Stateless multicast switching in software defined networks", 2016 IEEE International Conference on Communications (ICC), DOI 10.1109/icc.2016.7511036, May 2016,  
<<https://doi.org/10.1109/icc.2016.7511036>>.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981,  
<<https://www.rfc-editor.org/info/rfc791>>.

- [RFC1752] Bradner, S. and A. Mankin, "The Recommendation for the IP Next Generation Protocol", RFC 1752, DOI 10.17487/RFC1752, January 1995, <<https://www.rfc-editor.org/info/rfc1752>>.
- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <<https://www.rfc-editor.org/info/rfc1918>>.
- [RFC2009] Imielinski, T. and J. Navas, "GPS-Based Addressing and Routing", RFC 2009, DOI 10.17487/RFC2009, November 1996, <<https://www.rfc-editor.org/info/rfc2009>>.
- [RFC2101] Carpenter, B., Crowcroft, J., and Y. Rekhter, "IPv4 Address Behaviour Today", RFC 2101, DOI 10.17487/RFC2101, February 1997, <<https://www.rfc-editor.org/info/rfc2101>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, DOI 10.17487/RFC2663, August 1999, <<https://www.rfc-editor.org/info/rfc2663>>.
- [RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <<https://www.rfc-editor.org/info/rfc2865>>.
- [RFC3118] Droms, R., Ed. and W. Arbaugh, Ed., "Authentication for DHCP Messages", RFC 3118, DOI 10.17487/RFC3118, June 2001, <<https://www.rfc-editor.org/info/rfc3118>>.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<https://www.rfc-editor.org/info/rfc3972>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.

- [RFC4014] Droms, R. and J. Schnizlein, "Remote Authentication Dial-In User Service (RADIUS) Attributes Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Information Option", RFC 4014, DOI 10.17487/RFC4014, February 2005, <<https://www.rfc-editor.org/info/rfc4014>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4581] Bagnulo, M. and J. Arkko, "Cryptographically Generated Addresses (CGA) Extension Field Format", RFC 4581, DOI 10.17487/RFC4581, October 2006, <<https://www.rfc-editor.org/info/rfc4581>>.
- [RFC4982] Bagnulo, M. and J. Arkko, "Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs)", RFC 4982, DOI 10.17487/RFC4982, July 2007, <<https://www.rfc-editor.org/info/rfc4982>>.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, DOI 10.17487/RFC5533, June 2009, <<https://www.rfc-editor.org/info/rfc5533>>.
- [RFC5795] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObusT Header Compression (ROHC) Framework", RFC 5795, DOI 10.17487/RFC5795, March 2010, <<https://www.rfc-editor.org/info/rfc5795>>.
- [RFC6250] Thaler, D., "Evolution of the IP Model", RFC 6250, DOI 10.17487/RFC6250, May 2011, <<https://www.rfc-editor.org/info/rfc6250>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <<https://www.rfc-editor.org/info/rfc6275>>.
- [RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <<https://www.rfc-editor.org/info/rfc6282>>.

- [RFC6740] Atkinson, R.J. and S.N. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, DOI 10.17487/RFC6740, November 2012, <<https://www.rfc-editor.org/info/rfc6740>>.
- [RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <<https://www.rfc-editor.org/info/rfc7228>>.
- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <<https://www.rfc-editor.org/info/rfc7258>>.
- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC7400] Bormann, C., "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 7400, DOI 10.17487/RFC7400, November 2014, <<https://www.rfc-editor.org/info/rfc7400>>.
- [RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <<https://www.rfc-editor.org/info/rfc7401>>.
- [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>>.
- [RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <<https://www.rfc-editor.org/info/rfc7872>>.
- [RFC8060] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", RFC 8060, DOI 10.17487/RFC8060, February 2017, <<https://www.rfc-editor.org/info/rfc8060>>.



- [RFC8061] Farinacci, D. and B. Weis, "Locator/ID Separation Protocol (LISP) Data-Plane Confidentiality", RFC 8061, DOI 10.17487/RFC8061, February 2017, <<https://www.rfc-editor.org/info/rfc8061>>.
- [RFC8105] Mariager, P., Petersen, J., Ed., Shelby, Z., Van de Logt, M., and D. Barthel, "Transmission of IPv6 Packets over Digital Enhanced Cordless Telecommunications (DECT) Ultra Low Energy (ULE)", RFC 8105, DOI 10.17487/RFC8105, May 2017, <<https://www.rfc-editor.org/info/rfc8105>>.
- [RFC8163] Lynn, K., Ed., Martocci, J., Neilson, C., and S. Donaldson, "Transmission of IPv6 over Master-Slave/Token-Passing (MS/TP) Networks", RFC 8163, DOI 10.17487/RFC8163, May 2017, <<https://www.rfc-editor.org/info/rfc8163>>.
- [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>>.
- [RFC8280] ten Oever, N. and C. Cath, "Research into Human Rights Protocol Considerations", RFC 8280, DOI 10.17487/RFC8280, October 2017, <<https://www.rfc-editor.org/info/rfc8280>>.
- [RFC8376] Farrell, S., Ed., "Low-Power Wide Area Network (LPWAN) Overview", RFC 8376, DOI 10.17487/RFC8376, May 2018, <<https://www.rfc-editor.org/info/rfc8376>>.
- [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>>.
- [RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/info/rfc8484>>.
- [RFC8609] Mosko, M., Solis, I., and C. Wood, "Content-Centric Networking (CCNx) Messages in TLV Format", RFC 8609, DOI 10.17487/RFC8609, July 2019, <<https://www.rfc-editor.org/info/rfc8609>>.
- [RFC8724] Minaburo, A., Toutain, L., Gomez, C., Barthel, D., and JC. Zuniga, "SCHC: Generic Framework for Static Context Header Compression and Fragmentation", RFC 8724, DOI 10.17487/RFC8724, April 2020, <<https://www.rfc-editor.org/info/rfc8724>>.

- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.
- [RFC8926] Gross, J., Ed., Ganga, I., Ed., and T. Sridhar, Ed., "Geneve: Generic Network Virtualization Encapsulation", RFC 8926, DOI 10.17487/RFC8926, November 2020, <<https://www.rfc-editor.org/info/rfc8926>>.
- [RFC8939] Varga, B., Ed., Farkas, J., Berger, L., Fedyk, D., and S. Bryant, "Deterministic Networking (DetNet) Data Plane: IP", RFC 8939, DOI 10.17487/RFC8939, November 2020, <<https://www.rfc-editor.org/info/rfc8939>>.
- [RFC8981] Gont, F., Krishnan, S., Narten, T., and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6", RFC 8981, DOI 10.17487/RFC8981, February 2021, <<https://www.rfc-editor.org/info/rfc8981>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.
- [ROBUSTSDN] Canini, M., Kuznetsov, P., Levin, D., and S. Schmid, "A distributed and robust SDN control plane for transactional network updates", 2015 IEEE Conference on Computer Communications (INFOCOM), DOI 10.1109/infocom.2015.7218382, April 2015, <<https://doi.org/10.1109/infocom.2015.7218382>>.
- [ROHC] Fitzek, F., Rein, S., Seeling, P., and M. Reisslein, "RObust Header Compression (ROHC) Performance for Multimedia Transmission over 3G/4G Wireless Networks", Wireless Personal Communications Vol. 32, pp. 23-41, DOI 10.1007/s11277-005-7733-2, January 2005, <<https://doi.org/10.1007/s11277-005-7733-2>>.
- [SFCANYCAST] Wion, A., Bouet, M., Iannone, L., and V. Conan, "Distributed Function Chaining with Anycast Routing", Proceedings of the 2019 ACM Symposium on SDN Research, DOI 10.1145/3314148.3314355, April 2019, <<https://doi.org/10.1145/3314148.3314355>>.

- [SPHINX] Danezis, G. and I. Goldberg, "Sphinx: A Compact and Provably Secure Mix Format", 2009 30th IEEE Symposium on Security and Privacy, DOI 10.1109/sp.2009.15, May 2009, <<https://doi.org/10.1109/sp.2009.15>>.
- [TOR] "The Tor Project", n.d., <<https://www.torproject.org/>>.
- [TRANSACTIONSDN] Curic, M., Despotovic, Z., Hecker, A., and G. Carle, "Transactional Network Updates in SDN", 2018 European Conference on Networks and Communications (EuCNC), DOI 10.1109/eucnc.2018.8442793, June 2018, <<https://doi.org/10.1109/eucnc.2018.8442793>>.
- [UA-DHCP] Komori, T. and T. Saito, "The secure DHCP system with user authentication", 27th Annual IEEE Conference on Local Computer Networks, 2002. Proceedings. LCN 2002., DOI 10.1109/lcn.2002.1181774, n.d., <<https://doi.org/10.1109/lcn.2002.1181774>>.
- [VPN] Khanvilkar, S. and A. Khokhar, "Virtual private networks: an overview with performance evaluation", IEEE Communications Magazine Vol. 42, pp. 146-154, DOI 10.1109/mcom.2004.1341273, October 2004, <<https://doi.org/10.1109/mcom.2004.1341273>>.
- [WireGuard] Donenfeld, J., "WireGuard: Next Generation Kernel Network Tunnel", Proceedings 2017 Network and Distributed System Security Symposium, DOI 10.14722/ndss.2017.23160, 2017, <<https://doi.org/10.14722/ndss.2017.23160>>.

#### Acknowledgments

Thanks to all the people that shared insightful comments both privately to the authors as well as on various mailing list, especially on the INTArea Mailing List. Also thanks for the interesting discussions to Carsten Borman, Brian E. Carpenter.

#### Authors' Addresses

Yihao Jia  
Huawei Technologies Co., Ltd  
156 Beiqing Rd.  
Beijing  
100095  
P.R. China  
Email: [jiayihao@huawei.com](mailto:jiayihao@huawei.com)

Dirk Trossen  
Huawei Technologies Duesseldorf GmbH  
Riesstr. 25C  
80992 Munich  
Germany  
Email: dirk.trossen@huawei.com

Luigi Iannone  
Huawei Technologies France S.A.S.U.  
18, Quai du Point du Jour  
92100 Boulogne-Billancourt  
France  
Email: luigi.iannone@huawei.com

Paulo Mendes  
Airbus  
Willy-Messerschmitt Strasse 1  
81663 Munich  
Germany  
Email: paulo.mendes@airbus.com

Nirmala Shenoy  
Rochester Institute of Technology  
New-York, 14623  
United States of America  
Email: nxsvks@rit.edu

Laurent Toutain  
IMT-Atlantique  
2 rue de la Chataigneraie  
CS 17607  
35576 Cesson-Sevigne Cedex  
France  
Email: laurent.toutain@imt-atlantique.fr

Abraham Y. Chen  
Avinta Communications, Inc.  
142 N. Milpitas Blvd.  
Milpitas, CA, 95035-4401  
United States of America  
Email: AYChen@Avinta.com

Dino Farinacci  
lispers.net  
United States of America  
Email: farinacci@gmail.com