Independent Submission
Request for Comments: 8369
Category: Informational
ISSN: 2070-1721

H. Kaplan
128 Technology

1 April 2018

Internationalizing IPv6 Using 128-Bit Unicode

Abstract

It is clear that Unicode will eventually exhaust its supply of code points, and more will be needed. Assuming ISO and the Unicode Consortium follow the practices of the IETF, the next Unicode code point size will be 128 bits. This document describes how this future 128-bit Unicode can be leveraged to improve IPv6 adoption and finally bring internationalization support to IPv6.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8369.

Table of Contents

1. Introduction

Unicode [Unicode] is currently limited to 1,114,112 code points, encoded in various encoding formats (e.g., UTF-8, UTF-16, UTF-32). At the time of this document's publication, 136,755 code points have been allocated, with more already in the approval process. Every year, more writing scripts, symbols, and emojis are added, while none are removed. After consulting expert mathematicians, we have determined that the world will run out of code points someday in the future.

While it might appear that the current rate of code point allocation gives us plenty of time to deal with the exhaustion problem, the Internet's history has shown that popular number spaces do not fill up linearly, but rather exponentially. And once the size of a particular number space becomes entrenched, it takes decades to migrate to a larger one. Therefore, the code point number space must be increased as soon as possible.

The details for expanding the Unicode code point space are not covered in this document. Such details need to be worked out between the IETF, ISO, the Unicode Consortium, and various gods. We assume, however, that the code point space will need to grow dramatically, and there will continue to be a need for a fixed-length encoding scheme similar to UTF-32. Naturally, the next size increment should go from UTF-32 to UTF-128, and thus the rest of this document follows this assumption.

This new 128-bit Unicode code point space can be leveraged by the IETF to address one of the lingering issues with IPv6: there's not much left to standardize. With the changes described in this document, the IETF will be kept busy for decades to come. It also enables new features and market opportunities, to help the global economy. This in turn will increase tax revenues for governments, which eventually may lead to increased funds for combating global warming. Therefore, the ultimate goal of this document is to reduce global warming.

1.1. Requirements Language

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. All other words SHOULD be interpreted as described by the Oxford English Dictionary OED [OED], which MAY be considered almost as authoritative for word definitions as the IETF.

1.2. Definitions

UTF-128: A fixed-length encoding for 128-bit Unicode. It is implemented as an array of bytes in the same manner as legacy IPv6 addresses to avoid endianness issues.

Short-Name Tag: A short descriptive name for a Unicode character code point, surrounded by colons (:). For example ":garfield:" represents the Unicode code point for the Garfield cat imoji.

Emoji: Pictographic symbol encoded in Unicode, used to express a general item, concept, or emotion.

Imoji: Pictographic symbol encoded in Unicode, used to represent an individual, specific thing: a specific human, a favorite pet, a famous animal, etc.

Amoji: Pictographic symbol encoded in Unicode, used to represent an individual of an alien species.

Umoji: Pictographic symbol encoded in Unicode, used to represent unknown things not covered by the other mojis.

Omoji: Pictographic symbol encoded in Unicode, used to represent obfuscated identities, used as addresses for the purpose of privacy.

2. The Need for 128-Bit Code Points

The exponentially increasing demand for Unicode character code points might not be obvious at first glance. While it is true that the number of languages and their writing scripts do not grow quickly over time, one type of "character" will: emojis. Unicode has barely begun providing code points for all of the various emojis currently in use, and it is likely that more emojis will be created in the future. For example, there are still missing emoji symbols for most types of food and drink, the flags of each town and city on Earth, all human sporting and leisure activities including all local and national sports teams and players, and every plant and animal species and gender.

Furthermore, it has become common for some applications to allow their users to create custom emojis, whereby the user can provide the graphic to display for a new "character". For example, a user might set their chat application to display a graphic of Carlos Ramirez’s popular "Trollface" meme [TROLL], using the short-name tag ’:trollface:’ in their chat application. All other users of the same chat app will be able to see and use the same custom trollface emoji.

However, since there is no Unicode code point for :trollface:, the chat app cannot export the trollface emoji to other chat apps or protocols, such as Internet Relay Chat (IRC) or the Extensible Messaging and Presence Protocol (XMPP). This represents a clear interoperability issue.

In the future, it might also become desirable to assign each human a Unicode code point to represent them, similar to names, with the glyph being a picture of their face or a custom graphic. These personal code points are not truly "emojis" in the classical sense of being generic concepts, so we've decided to give them a new name to avoid confusion: imoji. Unlike human names, code points for imojis will be unique per human, for all space and time. Favorite pets and famous animals can also be assigned imojis.

Lastly, if we ever encounter sentient species from other planets, they too will need Unicode code points for their writing scripts and emojis; and they will each need unique amojis (imojis for aliens), for whatever form their individual identity might take. Section 4 of RFC 8136 [RFC8136] clearly supports such a scenario, with the new UFO IPv6 option.

Based on the above obvious use cases, it is clear that the current ~1 million code points are nowhere near enough. Increasing to 64 bits might be sufficient for now, but since this will be a painful transition process no matter the size, we believe jumping to 128 bits is the appropriate choice.

Note: The current limit of ~1 million code points is a formal limit due to what UTF-16 can encode today. Increasing the limit will either require deprecating UTF-16 or paying a hefty overhead penalty to encode 128 bits across many pairs of surrogate code points. Since the ultimate goal of this document is to reduce global warming, the challenge of choosing between deprecating UTF-16 or paying the overhead price is a trivial dilemma to solve by comparison.

3. Unicode IPv6 Addresses

Assuming the new Unicode code point space is 128 bits -- excluding some reserved bits for backwards compatibility and future expansion -- it seems only natural to use Unicode code points for IPv6 addresses, and vice versa. This leads to some exciting changes, such as:

- IPv6 addresses no longer need to be typed as hex values -- instead, the glyph for the script character, symbol, emoji, or imoji representing that address can be input by the user, and it will be displayed by the application as the graphic itself. From the user's perspective, this will also be more compact than the representation described in RFC 1924 [RFC1924].

- Network monitoring and troubleshooting tools can now display pretty glyphs in place of ugly IPv6 addresses, leading to less stress on the eyes of network administrators.

- For cases where graphical glyphs cannot be used, such as IETF documents, we can deprecate the legacy textual notation of IPv6 addresses of the style '2001:db8:85a3::8a2e:370:7334' to the simpler Unicode textual notation 'U+20010DB885A3000000008A2E03707334'. Using the short-name tag is also possible, such as ':v6address-1:'.

Due to the nature of having IPv6 addresses be Unicode code points, RFC 8135 [RFC8135] is made obsolete by this document. It was found to be too complex to implement anyway.

3.1. Reserved Addresses

Some address code points will be inappropriate for IPv6 addressing, such as formatting characters and control codes. Such code points MUST NOT be used for IPv6 addresses.

We do, however, still need to reserve some code points for private network use. Since no sentient life has been found on Mars, the code points that would have been allocated for Martian imojis are hereby allocated for this private use. These addresses are thus called "Martians", also known as "Bogons" due to them being bogus.

Note: Should life be found on Mars in the future, new code points will be allocated for them. To avoid confusion, they will be called "Barsoom Indigenous Glyph Off-world Network" addresses, or "Bigons" (pronounced "bye-gons"). We're certain the Martians will let Bogons be bygones, and Bigons be Bigons.

3.2. Multicast

In both IPv4 and IPv6, multicast addresses have been relegated to predefined IP address ranges, limiting how many multicast groups could be used simultaneously. Given the rise of broadcasting-style social media platforms, and the market demand for individuals to be watched/followed by numerous random strangers constantly, it seems clear that we need to be able to multicast everything, all the time.

To address this need, the high-order bit of the 128-bit code point space SHALL be reserved to indicate multicast. All valid code points (i.e., IPv6 addresses) will thus have multicast counterparts. For example, the code point allocated for :cat: is U+1F408. The multicast group U+80000000000000000000000000001F408 is thus for content about cats. Note that this is for general cat content -- other code points are allocated for specific cat content, such as joy cat, grinning cat, pouty cat, etc. For an individual cat like Garfield, setting the high-order bit to the code point allocated for :garfield: will indicate that it is multicast content about Garfield.

Source-specific multicast also plays a role; for example, joining the :garfield: multicast group and restricting it to a source of :garfield: results in only receiving content about Garfield, from Garfield.

3.3. IPv6 Routing

There should be little impact on routing using code-point-based IPv6 addresses. There might be some exponential growth in routing and forwarding tables due to difficulties in aggregating code points; hopefully, this will be offset by increases in processor and memory capacity. Of course this will also drive the need to frequently upgrade networking hardware, resulting in a boost to the global economy, and thus a reduction in global warming.

One improvement to routing that MAY be considered is for scenic routing as defined by RFC 7511 [RFC7511]. With emojis and imojis being available for addressing, we can now specify which exact type of scenery to visit along the way, or even which exact avian carrier [RFC6214] to ride with. Note that avian carriers as described in RFC 1149 [RFC1149] are not supported, since they only support IPv4.

4. Using Unicode IPv6 Addresses

4.1. Uniform Resource Identifiers

Uniform Resource Identifiers (URIs) and Uniform Resource Locators (URLs) already support Unicode through Internationalized Resource Identifiers (IRIs), but these are merely a means to use multiple Unicode characters to indicate a resource. With 128-bit Unicode, the number space is large enough to identify each resource with a single Unicode character. Why waste space and time typing out multiple characters, when you can just use one?

For URLs, this new model might only mean using a single Unicode character for the hostname portion -- for example, a corporate logo in place of the legacy corporate domain name. Another alternative is to allocate a code point for the entire host and path, possibly even including the scheme. These types of decisions can be made in future IETF Working Groups.

The interesting aspect of this change for URIs/URLs is that no address lookup needs to be performed. The single 128-bit Unicode for the URL *is* the IPv6 address. An additional step is only needed if the user inputs a private Unicode character or short-name tag that needs to be converted to a publicly allocated one. This would require Network Address Translation (NAT) from the private code point or short-name tag to a public Unicode code point. This can be done locally, thus finally bringing NATs into the last part of the Internet in which they are not currently deployed: the user’s application.

4.2. Address Allocation and Resolution

It is obvious that once a single 128-bit Unicode character is used for addresses and URIs, using domain names will quickly become obsolete. The subsequent collapse of the domain name industry presents a threat to the world economy, which MUST be addressed.

One solution to this danger is to establish a Unicode registry model and an accompanying Code Point Unicode Resolution System (CPURS, pronounced "keepers"). CPURS would replace DNS and provide an architecture and resolution mechanism to resolve Unicode code points to their registered glyphs and short-name tags, and vice versa. The new Unicode registries and registrars would thus replace the legacy domain name counterparts. This would lead to a new gold rush for registering Unicode code points for corporate logos and product icons, and thus usher in an era of economic prosperity, which would eventually reduce global warming.

Once Unicode registries and CPURS are in place, IPv6 addresses would be allocated by registering code points through that system; they would no longer be registered by IANA and RIRs. This is not a major concern, however, because any tax revenue loss will be more than offset by Unicode registries allocating code points. Furthermore, in order to make CPURS possible, the actual graphic files for the glyphs need to be standardized and created in numerous formats and sizes, with various intellectual property rules. This will provide more work for graphic artists and lawyers, further increasing tax revenue.

The astute reader might ask why we need CPURS if Unicode translation is performed locally on hosts today. The answer is volume: it is unlikely that host applications can keep up with the rate of new Unicode code points being allocated for emojis, imojis, and umojis. While application and operating system updates have been occurring at an ever-increasing rate, and will soon reach the same rate as human births, it is doubtful that it will ever reach the rate of sentient extraterrestrial births. Therefore, we need a system that can scale to reach such volume before we make first contact; otherwise, the diplomatic failure to quickly provide the aliens with amojis of their own may lead to armed conflict. An armed conflict with other sentient beings capable of reaching Earth might increase global warming, defeating this document’s ultimate purpose.

5. Summary

There is still much to be decided on, most of which is frankly rather boring. It is clear, however, that 128-bit Unicode code points will be needed eventually, and IPv6 addressing MUST be migrated to it. Thus, the time to act is now!

6. IANA Considerations

This document has no IANA actions.

7. Security Considerations

The main security concern with using 128-bit Unicode for IPv6 addressing is the need for privacy, in terms of anonymity. If an IPv6 packet is sent with an imoji or amoji address, then man-in-the-middle devices in the network will know the specific human or alien that sent or received the packet. Using such information might lead to heated discussions, thereby increasing global warming.

To address this concern, an IPv6 address MAY be obfuscated by using an omoji. An omoji is simply the original Unicode code point but with the least-significant bit set; all other types of 128-bit Unicode code points MUST have the least-significant bit cleared. The

graphical representation of an omoji is the same as its unobfsucated moji counterpart, except that it is covered over by a solid black block.

By setting the least-significant bit of the source or destination and thus turning it into an omoji, the IPv6 address is obfuscated and the true identity cannot be determined, while IPv6 routers can still route the packet appropriately. Note that this only provides a bit of privacy, but every bit helps.

8. References

8.1. Normative References

[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>.

8.2. Informative References

[OED] Oxford University Press, "Oxford English Dictionary", <http://www.oed.com>.

[RFC1149] Waitzman, D., "Standard for the transmission of IP datagrams on avian carriers", RFC 1149, DOI 10.17487/RFC1149, April 1990, <https://www.rfc-editor.org/info/rfc1149>.

[RFC1924] Elz, R., "A Compact Representation of IPv6 Addresses", RFC 1924, DOI 10.17487/RFC1924, April 1996, <https://www.rfc-editor.org/info/rfc1924>.

[RFC6214] Carpenter, B. and R. Hinden, "Adaptation of RFC 1149 for IPv6", RFC 6214, DOI 10.17487/RFC6214, April 2011, <https://www.rfc-editor.org/info/rfc6214>.

[RFC7511] Wilhelm, M., "Scenic Routing for IPv6", RFC 7511, DOI 10.17487/RFC7511, April 2015, <https://www.rfc-editor.org/info/rfc7511>.

[RFC8135] Danielson, M. and M. Nilsson, "Complex Addressing in IPv6", RFC 8135, DOI 10.17487/RFC8135, April 2017, <https://www.rfc-editor.org/info/rfc8135>.

[RFC8136] Carpenter, B. and R. Hinden, "Additional Transition Functionality for IPv6", RFC 8136, DOI 10.17487/RFC8136, April 2017, <https://www.rfc-editor.org/info/rfc8136>.

[TROLL] The Meme Wikia, "Trollface", <http://meme.wikia.com/wiki/Rule_63?oldid=23602>.

[Unicode] The Unicode Consortium, "Unicode", <http://unicode.org>.

Acknowledgements

The authors wish to thank the following people for providing the inspiration for this work: Cal Henderson, Carlos Ramirez, Graham Linehan, Agnetha Faltskog, Bjorn Ulvaeus, Benny Andersson, and Anni-Frid Lyngstad.

Author’s Address

Hadriel Kaplan
128 Technology
Burlington, MA
United States of America

Email: hadriel@128technology.com