INTERNET-DRAFT
"Internet Protocol Five Fields - Design Decisions",
Alexey Eromenko, 2016-09-29,
<draft-eromenko-ipff-design-decisions-00.txt>
expiration date: 2017-03-29

Intended status: Informational

A.Eromenko
September 2016

Internet Protocol "Five Fields": Design Decisions

Abstract

Goal of IP-FF: provide future growth, without design complexity of IPv6. This document writes the design decisions behind IP-FF and explains why they were done.

Status of This Memo

Copyright Notice

1. Overall architecture based on IPv4 (for easy code and network migration, and easy understanding by developers and network engineers)

!!!

1.2. Address space was designed to be big enough for the next several hundred years *but* _optimized for human memory_, rather than computer memory.
It turned out that 50-bit addresses are very good at it.
(computer memory is cheap nowadays, even if we use a 64-bit data field for only 50-bits of actual data).

1.3. ARP stays; It's easy to understand and won't require massive network configuration changes.

1.4. Links are required to support 1280 bytes MTU (I expect physical networks to be compatible with IPv6 by this time. which mandates this size)

2. Port numbers stay 16-bit. While I considered 32-bit, I rejected this idea.

2.1. will push carriers towards carrier-grade NATs (CGNs), which is a non-goal.

2.2. extra overhead of 4 bytes per packet. (0.27% slower for 1500 byte packets)

2.3. I was unable to find major advantages.

3. Ports were moved from Layer 4 to Layer 3 header; which allows for:

3.1. Flow-based routing (via 5-tuple or 6-tuple rule).

3.2. Faster Firewalls

3.3. Simpler fragmentation for FDP

3.4. A bit higher overhead for protocols, that do not use 16-bit ports. (a minor evil, but still.)

4. UDP protocol length field removal:

4.1. Is not needed, as TCP lacks length field, and does fine.

4.2. Allows for jumbo frames.

4.3. Allows for a stronger checksum.

5. TCP-64-bit:

5.1. needed for faster speeds (1 Terabit or more - I envision for end-user devices, at Earth distances. Lots of data in mid-air, unacknowledged).

5.2. Ensuring reliability at such speeds requires *much* stronger checksums.

* TCP protocol originally designed to guarantee reliability, and it's 16-bit checksum worked fine in the 80's and 90's with 56 kbit/s WAN speeds, but is not adequate nowadays at gigabit speeds let alone future-networks at terabit speeds. A stronger 64-bit checksum restores this original guarantee.

6. Type of Service stays for compatibility reasons

7. TTL/HTL (Hops-to-Live) was extended mostly due to my envision of future

network virtualization, where virtual routers and containers (network namespaces) speak to each other.
For physical networks TTL of 255 I expect it to be enough for this century.

8. **Payload length was reduced to 14-bits, because it's enough to handle both Ethernet Jumbo frames (9 KB) as well as WiFi frames (8 KB).**

8.a. A standard IP-FF extension provided for Jumbograms (4 MiB size IPFF packets), just-in-case.

9. **Compatibility: IP-FF networks are theoretically backwards compatible with IPv4 networks over a NAT router.**

10. **Extensions mechanism was completely rewritten for simplicity (of** understanding).

11. **Fragmentation: moved to Layer 4.**
* TCP doesn't need a fragmentation, due to sliding windows.
* UDP in IP-FF has a special version called FDP -- Fragmented Datagram Protocol.
It's fragmentation has moved to layer 4, checksums are for a single fragment, rather than full packet, -and- port is visible at layer 3, so NAT with fragmented packets should be *much* easier with IP-FF.

Fragmentation theory:
Basically here are few possibilities:

1. **Fragmentation-and-reassembly at every hop. (I don't know if anybody implements it)**
2. **IPv4-style-fragmentation -- fragmentation per every hop, reassembly at destination end.**
3. **IPv6-style-fragmentation -- fragmentation only at source end,** reassembly at destination end.
4. **No fragmentation at all (the advantage here: faster Router processing vs #1 or #2 and less implementation bugs);** Assuming standard packet size is defined at 1280 bytes, like in IPv6
5. **MTU path discovery via ICMP -- RFC-1981**
6. **MTU path discovery via TCP (or other Transport) -- RFC-4821** (or another way)

I'm leaning towards 3 + 4 + 6 solution in my own protocol, IP-FF.
FDP is basically IPv6 style fragmentation for UDP, but improved, while TCP doesn't need it at all.

12. **DNS and DHCP went through minimal changes, required for IP-FF.**
* DNS resembled more a DNS version from IPv6, while DHCP is a reworked version from IPv4 DHCP.

13. **ARP protocol got a new feature:**
* IP-FF session ID -- a protection vs. duplicate MAC address (common in virtualized environments)

14. ICMP changes:

```
Destination Unreachable Message (Type = 1) (IPv4-style)
      2 = protocol unreachable;
vs.
Parameter Problem Message (Type = 4) (IPv6 style)
      1 = Unrecognized Next Header type encountered
```

I'm leaning towards IPv4 style, because only the end-node can tell if the "protocol is unreachable".

Authors' Contacts

Alexey Eromenko
Israel

Skype: Fenix_NBK_
EMail: al4321@gmail.com
Facebook: https://www.facebook.com/technologov