IP over CAN[*]

Christoffer Holmstedt
christoffer.holmstedt@gmail.com

ABSTRACT

The Internet Protocol (IP) is a connection oriented communication protocol based on the idea that all hosts have an address and communication is done by end-to-end communication. The Controller Area Network (CAN) is a message oriented protocol where all hosts connected to a single CAN bus can read everything and often write back to.

As the world moves towards IP based communication for Internet of Things it's worth looking into if IP can be adapted to work across a CAN bus. This report goes through what has been done in the Naiad project and what future work should be undertaken. In the end IP over CAN was not used in the final design of the Naiad AUV.

1. INTRODUCTION

As part of the Space plug-and-play Avionics requirement focused on IP over CAN. The following sections will give a short introduction to the Internet Protocol (IP) and the Controller Area Network (CAN or CAN bus).

1.1 Internet Protocol (IP)

The Internet Protocol is the main protocol for addressing hosts on a single, or multipe link networks. Internet Protocol version 4 is the version currently most hosts use but with no more addresses available for new devices a move to Internet Protocol version 6 is currently ongoing [21].

1.1.1 IP version 4 (IPv4)

IPv4, originally defined as a standard in September 1981 [6], has seen a few updates throughout the years. The core part is that each host connected to a network has a 32 bit address. For readability this address is often written in decimal form as four octets (e.g. 192.168.0.1).

As the number of connected devices grew more than expected in the 1990's "Network Address Translation" (NAT) was introduced to limit the number of publically accessible IP addresses assigned and at the same time improve routing for Internet Service Providers [7, 9, 13].

One of the most common ways to assign IP addresses on a network is through the Dynamic Host Configuration Protocol (DHCP) [10, 11, 17]. When a host connects to the network, the client sends a DHCP request and the DHCP server(s) available on the network replies back with a DHCP offer containing a free IP address the new host can use. As identification the client uses either its MAC Address or another client's identification value depending one the version of the protocol that is used. After this process end-to-end communication is possible within the network.

1.1.2 IP version 6 (IPv6)

IPv6 was first specified as proposed standard RFC 1883 [8] in December 1995 and now obsoleted by RFC 2460 [12]. The motiviation for its creation is that it was in the end of 1980's apparent that the IPv4 address space was too small. Instead of the 32 bits addresses in IPv4, an IPv6 address consists of 128 bits. For readability this address is written in hexadecimal form in 8 groups separated by colons (e.g. 2a00:860:340:aabf:aabf:baad:3423:495).

RFC 4291 defines the IPv6 addressing architecture [16] while RFC 5952 [20] defines recommended text representation of IPv6 addresses.

In comparison with IPv4 the address assignment can be done in more than two different ways. The first way is by static assignemt which is equal to IPv4. The second way is DHCPv6 that works in a similiar manner to DHCP [14]. As a configuration option of DHCPv6, a third assignment procedure is possible with "DHCP Prefix Delegation" (DHCP-PD) [15] that assigns complete blocks of addresses to routers which are then free to distribute addresses within that block of addresses. If DHCPv6 isn't available "Stateless Address Autoconfiguration" (SLAAC) is a way for each host to generate and set a unique address for each of its interfaces. This also includes what is called "Duplicate Address Detection" to prevent multiple hosts from using the same IPv6 address [18, 19].

As more and more devices get connected to the internet, the "Internet of things" become a reality. For "Internet of things"

[*]This report was written during the fall of 2013 in an advanced level project course at Mälardalen University, Sweden.

most focus has been put on wireless technologies such as the 6LoWPAN [1] work and ROLL [22] within the IETF.

In April 2012 the IETF released a Best Current Practice (BCP) document "IPv6 Support Required for All IP-Capable Nodes". This BCP recommends that IPv6 must be implemented in all hosts that are IP capable [21].

1.2 Controller Area Network (CAN)

The CAN protocol is a message based broadcast protocol developed and specified by ROBERT BOSCH GmbH (Bosch). CAN Specification 2.0 was released in September 1991 and consists of two parts, A and B. An implementation of the CAN Specification must comply with either of the two parts or both. The main difference is the length of message IDs which in the standard format is 11 bits long and in the extended format is 29 bits long. Part A requires support for the standard format while Part B implementations must support standard format and extended format [3].

The key part of the CAN bus protocol is that all hosts connected to the same bus will be able to read simultaneously from the bus, it's up to each host to filter in the wanted messages. The filtering process is done by specifying one or multiple masks on each host that is used when reading incoming message IDs.

One important aspect is that if multiple hosts transmit a message at the same time with the same message ID but with different payloads the CAN bus will enter an error state. Multiple hosts are therefore not allowed to transmit messages with the same message ID.

CAN with Flexible Data-Rate (CAN FD) is a new specification which increase the data rate from a maximum of 1Mbit/s to 8Mbit/s [4].

1.3 IP over CAN

Previous work in this field is limited. In 2001 Petr Cach and Petr Fiedler created a first draft for IP over CAN [2]. They clearly state that their solution is not for hard real-time operations. In 2003 Ditze et. al. [5] took another approach with a larger prioritisation band.

The following parts of the report is structured as follows. Section 2 goes through what has been done with IP over CAN in the Naiad project and section 3 focuses on lessons learned and future work.

2. IMPLEMENTATION

The latest solution presented for IP over CAN is from Ditze et. al. with a 10 bits prioritisation field, 3 bits message type field, 8 bits destination address and 8 bits sender address. This solution is shown in table 1. It's from this solution work started, in the Naiad project.

Table 1: Mapping of message ID to IP over CAN bus.

Prioritisation	Msg	Sender Address	Dest. Address
10 bits	3 bits	8 bits	8 bits

In their solution Ditze et al. suggest a gateway that routes traffic to and from the CAN bus network. The routing that has to be done would map full 32 bit IP addresses to CAN bus IP addresses of 8 bits. This would be done by dropping the first 24 bits of the IPv4 address and only using the last 8 bits over the CAN Bus.

Ditze et al. don't go into detail about assignment of IP addresses but instead mention that it can be done statically or dynamically. As a goal for the Naiad project was to to use Space plug-and-play Avionics technology it introduce the requirement that low level address assignment had to be done dynamically, to support plug-and-play features. A first suggestion was to use a well-known message ID that all new nodes would transmit a DHCP request with that message ID. As the payload for the DHCP request, a host's identifier would be set. The problem with this approach is that when multiple hosts boot up simultaneously, a collision will occur and the CAN bus will go into an error state.

The next approach was to use the destination address field and sender address field with a total of 16 bits from the message ID as a unique host identifier field when sending a DHCP request message (shown in table 2). This approach seems to be viable and would give 16 bit addresses which might be too few to prevent accidental CAN bus collisions during boot up. The 16 bit identifier would have to be generated from some kind of hardware ID such as the MAC Address for ethernet devices.

Table 2: Mapping of message ID to IP over CAN bus with Host identifier during DHCP process.

Prioritisation	Msg	Host identifier
10 bits	3 bits	16 bits

When worked had reached this point other priorities were made in the project so no further progress was made.

3. CONCLUSION

The approach taken during the Naiad project started from the solution presented by Ditze et al. This approach seemed to have some good features to it. Among others the 10 bits prioritisation band seemed to be useful for systems with hard real-time requirements. In the end it turned out that the same prioritisation band might be too big which limits the number of bits that can be used of host identification during the address assignment procedure. Future work should look into the possibilty of using dynamic address assignment specified by Cach et al. while trying to keep the prioritisation band as big as possible.

At the time of writing the latest solution in this field is over 10 years old and puts focus on IPv4. As of 2012 the IETF recommends that all IP capable devices use IPv6. This recommendation should be followed and future work should start from a IPv6 point of view.

Current focus around IPv6 is often targeting wireless communication which is far from the CAN bus specification but some parts are similiar. The targetted wireless networks often run with small embedded 8-bit microcontrollers which is often the case for hosts connected to CAN busses as well.

Future work should take a closer look if any of the ideas behind protocols specified by 6LoWPAN working group or the ROLL Working Group in IETF can be used for IP over CAN.

4. REFERENCES

[1] 6lowpan working group charter. http://datatracker.ietf.org/wg/6lowpan/charter/. Accessed 2013-12-06.

[2] P. Cach and P. Fiedler. Ietf draft - ip over can. http://datatracker.ietf.org/doc/draft-cafi-can-ip/. Accessed 2013-12-02.

[3] Can bus 2.0 specification. http://www.bosch-semiconductors.de/media/pdf_1/canliteratur/can2spec.pdf. Accessed 2013-12-02.

[4] Can with flexible data-rate. http://www.bosch-semiconductors.de/media/pdf_1/canliteratur/can_fd_spec.pdf. Accessed 2013-12-06.

[5] M. Ditze, R. Bernhardi, G. Kämper, and P. Altenband. Porting the internet protocol to the controller area network. http://www.hurray.isep.ipp.pt/rtlia2003/full_papers/8_rtlia.pdf. Accessed 2013-12-03.

[6] Rfc 791, internet protocol. http://datatracker.ietf.org/doc/rfc791/. Accessed 2013-12-04.

[7] Rfc 1631, the ip network address translator (nat). http://datatracker.ietf.org/doc/rfc1631/. Accessed 2013-12-04.

[8] Rfc 1883, internet protocol, version 6 (ipv6) specification. http://datatracker.ietf.org/doc/rfc1883/. Accessed 2013-12-05.

[9] Rfc 1918 (bcp 5), address allocation for private internets. http://datatracker.ietf.org/doc/rfc1918/. Accessed 2013-12-04.

[10] Rfc 2131, dynamic host configuration protocol. http://datatracker.ietf.org/doc/rfc2131/. Accessed 2013-12-04.

[11] Rfc 2132, dhcp options and bootp vendor extensions. http://datatracker.ietf.org/doc/rfc2132/. Accessed 2013-12-04.

[12] Rfc 2460, internet protocol, version 6 (ipv6) specification. http://datatracker.ietf.org/doc/rfc2460/. Accessed 2013-12-05.

[13] Rfc 3022, traditional ip network address translator (traditional nat). http://datatracker.ietf.org/doc/rfc3022/. Accessed 2013-12-04.

[14] Rfc 3315, dynamic host configuration protocol for ipv6 (dhcpv6). http://datatracker.ietf.org/doc/rfc3315/. Accessed 2013-12-05.

[15] Rfc 3633, ipv6 prefix options for dynamic host configuration protocol (dhcp) version 6. http://datatracker.ietf.org/doc/rfc3633/. Accessed 2013-12-06.

[16] Rfc 4291, ip version 6 addressing architecture. http://datatracker.ietf.org/doc/rfc4291/. Accessed 2013-12-05.

[17] Rfc 4361, node-specific client identifiers for dynamic host configuration protocol version four (dhcpv4). http://datatracker.ietf.org/doc/rfc4361/. Accessed 2013-12-04.

[18] Rfc 4862, ipv6 stateless address autoconfiguration. http://datatracker.ietf.org/doc/rfc4862/. Accessed 2013-12-05.

[19] Rfc 4941, privacy extensions for stateless address autoconfiguration in ipv6. http://datatracker.ietf.org/doc/rfc4941/. Accessed 2013-12-05.

[20] Rfc 5952, a recommendation for ipv6 address text representation. http://datatracker.ietf.org/doc/rfc5952/. Accessed 2013-12-05.

[21] Rfc 6540, ipv6 support required for all ip-capable nodes. http://datatracker.ietf.org/doc/rfc6540/. Accessed 2013-12-06.

[22] Roll working group charter. http://datatracker.ietf.org/wg/roll/charter/. Accessed 2013-12-06.