LwIP Working Group
Internet-Draft
Intended status: Informational
Expires: April 27, 2011

H. Deng
China Mobile
S. Sakane
Cisco
W. Haddad
Ericsson
N. Kong
CNNIC

October 24, 2010

Problem Statement of Lightweight IP Protocols Design
draft-deng-lwip-ps-01

Abstract

Since small devices such as sensors are often required to be physically small and inexpensive, an implementation of the Internet protocols will have to deal with having limited computing resources and memory. This report describes the design and implementation of a small TCP/IP stack called lwIP that is small enough to be used in minimal systems.

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 http://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 April 27, 2011.

Table of Contents

1. Introduction

Technologies are evolving, but we keep inventing smaller things. For most devices that are commonly used in the Internet of Things applications, they are resource constrained. First, they are built on top of the constrained computing platform. e.g. with 8-bit microcontrollers and limited RAM and ROM. Secondly, the connectivity between the nodes and the outside network is constrained e.g, some networks go down to 20kbps and with limited delivery probability. Thirdly, these devices are energy constrained. They only have battery supply and the application users normally do not change their battery for long periods of time.

On the other hand, the applications require more and more resources on the the devices. In home network applications, the small monitoring devices are required to send vedio streamings to the host's mobile phones, and also in factory monitoring scenarios, the requirments for high data rate transmission is emphasized.

Given the above facts, IP protocols should be designed in as lightweight way as possible. This document summarizes those issues met by current practices. The document also introduces the current practices on implementing TCP/IP protocols in a lightweight way, and the problems met by implementers.

1.1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL","SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Usage Scenario

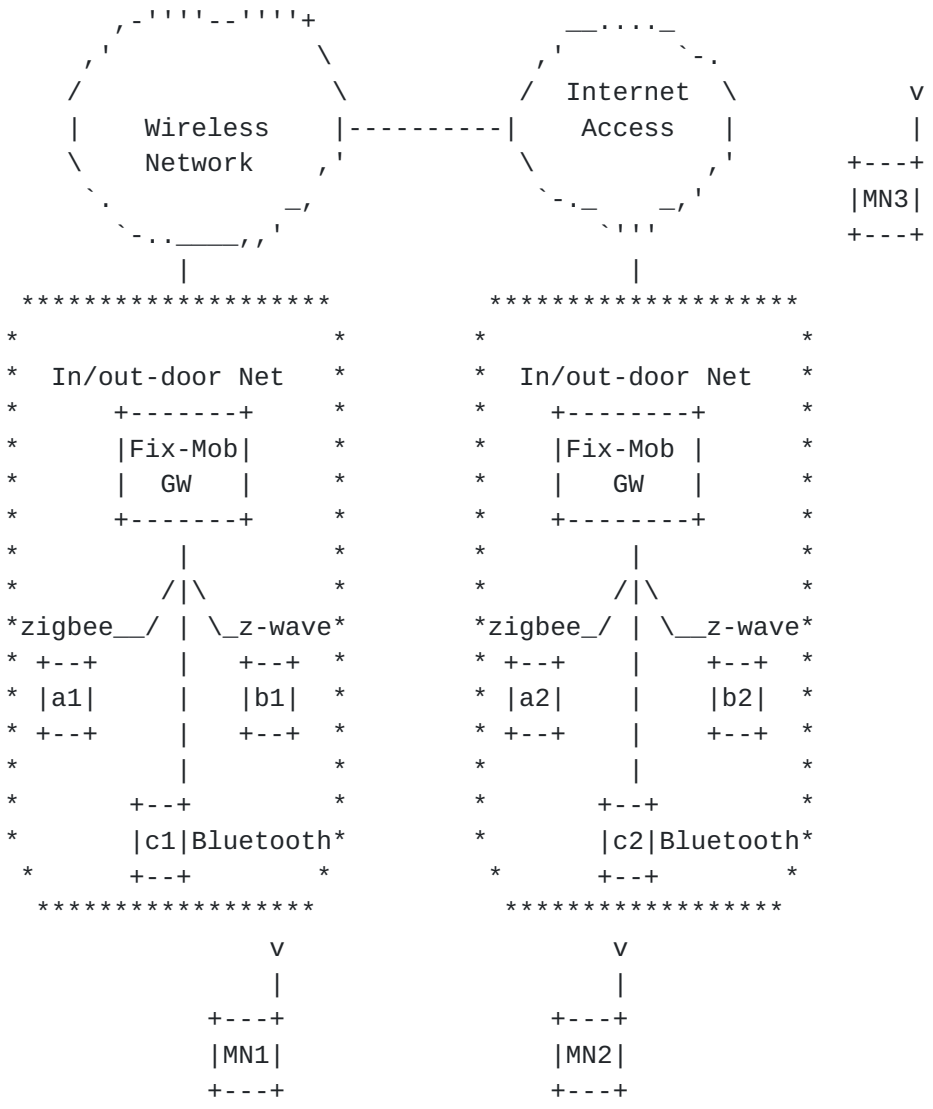

Figure 1: The Scenario

Currently many applications are built on this scenario. For example, a sensor for the intrusion detection sends messages to the host's mobile terminal (MN3) while they away from the house. The host can also send messages via the home gateway to home sensors in order to execute some commands, for instance, open the air conditioner before going home. For service discovery, the mobile node should be able to discover the service available nearby. For example, when a MN moves to a new environment (moving from its household to its work place), it should be able to discover the temperature and humidity information service nearby.

The connection between the fix-mobile gateway and different types of sensors is based on various wireless technologies, for example, IEEE 802.15.4, Bluetooth, Z-Wave and RFID. IP protocol, as an adapting layer, should run over all these technologies without loosing features of communicating, qos, security and etc. So in order to accommodate those different sensing devices, it is highly desirable for all the components to talk with IP. However, both the sensing devices and the mobile gateway are built on top of the constrained platform. They are basically constrained in three aspects:

- Energy Constrained. Most of the sensors do not have sustained power supply. Some of the gateway nodes also do not have power supply. Among the operations done by these nodes, communication is the most energy consuming. For scenarios like agriculture where the GW is most likely sitting somewhere outside in the field or floating on water without permanent power connectivity, the gateway are mostly out of sustained power supply. As a consequence, network stack implementations should be highly aware of these features.

- Computation Constrained. The sensor nodes are built upon 8-bit MCUs with server kilos of RAM. The gateway nodes, even if they are more powerful than sensors, are not fully capable due to economic reasons. They should maintain the upward connection with the global internal as well as the downward connection between the sensor nodes. In many deploy scenarios, those gateway nodes are built upon the cellular phone platform (ARM7 or ARM9 with reduced computation functionality). In these scenarios, implementations should be aware of the computational capability on the nodes. And if secure computations are involved, implementations will met more challenges.

- Network Constrained. The lower layer connection between the mobile gateway and the sensors is running at low data rate. For example, IEEE 802.15.4 provides over-the-air rates from 20kbps to 100kbps at 868/915 MHz. Implementations should be aware of the lower layer capability in order to be compliant.

3. Constraint and Requirements

3.1. Low Energy Consumption

This requirement is derived from several reasons.

- it is necessary to use the devices at long time. the device is sometimes driven with a non-rechargeable and non-replaceable battery.
- it is necessary to use the devices in the explosion atomosphere of the process automation.
- in the human sensing, in particular, the inside of the human body sensing, the battery is very low power.
- essentially, in the world wide movement, minimizing energy consumption must be considered.

3.2. Limited Memory Size

The sensors are running at very constrained platforms with 8-bit micro-controllers and several kilos RAM. Even the fixed/mobile gateways are not fully capable devices; they are built on cheap chipsets for economic reasons. It is required for applications to be deployed and developed above these constrained platforms.

3.3. Various Data Rates

It depends on the application requirement, video, audio streaming, or periodical sensing. The media type restricts the data rate. Actually, the technology develops the rate of the media. However, we have to consider about it for practical.

3.4. Rapid Deployment

Meanwhile, there are numerous kinds of mobile terminals with heterogeneous platforms. It is difficult for developers to design universal expansion hardware to help mobile terminals access the sensor networks. Consequently, it is desirable to see some rapid deployment techniques to faciliate the fix-mobile gateway accessing the sensor network without affecting the mobile terminal platforms too much. uSD [usd] is such a practice.

4. Current Implementations

4.1. uIP

uIP is an implementation of the TCP/IP protocol stack intended for small 8-bit and 16-bit microcontrollers. It is completely RFC1122 compliant but has some limitations. For instance, a retransmit is managed by the stack, but the data that needs to be retransmitted is requested from the user application.

uIP can be used together with Contiki, a very small OS which supports dynamic application download and a gui using VNC. The uIP stack uses less then 10kB ROM and 2kB RAM and Contiki can easily fit in 100kB ROM and 10kB RAM. You can use it any way you want as long as you leave a copy of the copyright notice in the source and/or documentation. uIP can be accessed via http://www.sics.se/~adam/uip/

4.2. LwIP

LwIP is a TCP/IP implementation designed for small code size and efficient memory usage. It is still widely used, and implemented. And is designed to use with or without an operating system. lwIP uses around 40kB of RAM and 30kB ROM and you can use it any way you want as long as you leave a copy of the copyright notice in the source and/or documentation. LwIP can be found via http://www.sics.se/~adam/lwip/

4.3. uC/IP

uC/IP is a TCP/IP stack developed for microcontrollers and embedded systems but is not often used. It based on the BSD TCP/IP implementations and is still a bit large compared to other implementations. uC/IP carries the BSD license so you can freely use it as long as you leave a copy of the copyright notice in the source and/or documentation. uC/IP can be found via http://ucip.sourceforge.net/

4.4. Blip on TinyOS

BLIP, the Berkeley Low-power IP stack, is an implementation in tinyos [tinyos] of a number of IP-based protocols. Using blip/tinyos, you will be able to form multi-hop IP networks consisting of different motes communicating over shared protocols. It has been tested on micaz, telosb, and epic platforms. It is not based on tinyos' active message layer, and the blip router is only supported on Linux, as of 2.1.1.

BLIP can be found via

http://smote.cs.berkeley.edu:8000/tracenv/wiki/blip.

4.5. TinyTCP

TinyTCP is a small implementation of the TCP and IP protocols, suitable for burning into ROM. A timer and an ethernet board are assumed. The implementation is based on busy-waiting, but the tcp_handler procedure could easily be integrated into an interrupt driven scheme. The TCP does not implement urgent pointers (easy to add), and discards segments that are received out of order. It ignores the received window and always offers a fixed window size on input (i.e., it is not flow controlled).

TinyTCP can be found via
http://www.unusualresearch.com/tinytcp/tinytcp.htm

5. Lightweight IP Protocols Implementation Issues

5.1. P1: Modularity and Layering

In many TCP/IP implementations, the layered protocol design has served as a guide. Each protocol shall be implemented separately from the other. However, implementing the protocols in a strictly layered way sometimes leads to a situiation where the communication overhead between the protocol layers degrades the overall performance [RFC0817]. To tackle this problem, care must be taken so that only the important information is shared among layers.

The other fact is that the operation systems used in the constrained systems most often do not maintain a strict protection barrier between the kernel and application processes, such as TinyOS and Contiki. This offers an opportunity for more relaxed communication between application and the lower layer protocols via shared memory, so that the internet-working stack can use the memory more efficiently.

So in order to improve performance in terms of processing speed and memory usage, implementors should in many situations violate the layering assumptions. For example, when verifying the checksum of an incoming TCP segment and when demultiplexing a segment, passing the source and destinations addresses to TCP via funtion calls will inevitably result into more memory consumption. In this case, making the TCP module aware of the structure of the IP header is more efficient althout it violates certain layering and modularity assumptions.

5.2. P2: Memory Usage Constraints

Constrained devices with several kilos of memory cannot service dynamic memeory allocation schemes. Most used MTU for the IP networks are 1500 bytes or bigger. One full size IPv4/IPv6 packet scales to more than a thousand bytes memory when using dynamic allocation. How to holding the incoming and outgoing packets to reduce the memory usage is a challenging question then.

For example, uIP [uip] uses preallocated buffers, in which a fixed table is used to hold the connection state while a global buffer is used to hold all the incoming and outgoing packets. Moreover, uIP also implements zero-copy mechanisms over the packet buffer to further reduce memory usage and minimize the data transfers between the TCP/IP stack and the application program. As a result, the size of the global packet buffer is determined by a configuration option at compile time, which guarantees that the buffer is large enough to accommodate a packet of the maximum allowed size. Nevertheless, due

to the implemented zero-copy mechanisms the application program must be carefully designed and pay special attention to all the packets arriving from the network to avoid packet loss, which occurs whenever the network device buffers become full.

5.3. P3: Inefficient Socket APIs

The BSD socket API requires the support of a multitasking system which imposes a significant overhead due to the need for task management, context switching and allocation of stack space. This is overbearing on many constrained platforms.

As a result, it is desirable to have a reduced event driven API with only the minimal and necessary functions is used to implement a TCP/IP stack optimized for small 8 and 16-bit systems. Using this approach the application program is only invoked in response to specific events, such as data arriving from a connection or an incoming connection request, which allows achieving minimum response times even in low-end systems

5.4. P4: Protocol Interoperability

Many implementations on constrained devices have implemeted different sets of features of the IP protocols. For example, some do not handle IP fragmentation and reassembly, and some do not handle IP options. Due to the lack of implementation guidelines and protocol implementation profiles, the interoperability between different implementations potentially is problem.

6. Other Issues

6.1. Lightweight IP Mobility Protocol

Mobility patterns for the small devices is different from what's been investigated in typical Internet IP mobility. IP mobility solutions, such as MIPv4/v6, DSMIPv6 or PMIPv6, normally have incurred overhead through the use of heartbeat signalings, IP tunnels and security associations. On small and constrained devices, they may not surrive, heartbeat signalings waking the device up too much for energy consumption. for example, Mobile IPv6 is too heavy in terms of security requirements and signaling exchange to be implemented on mobile GWs.

6.2. Lightweight Service Discovery

There are lots of constrained devices in the networks. To find a service of an application, a service discovery is required. There are several kind of the service discovery protocol like DNS-SD, SSDP, or XMPP. These protocols were not designed to assume running on the constrained devices. For example, XMPP is based on XML. It is indeed scalable to its contants. But, it is likely not "thing-friendly". Each of them are necessary to be considered whether it is suitable or not. Further investigation is needed.

6.3. Lightweight Global Name Services

Global name services such as DNS (Domain Name System) [RFC1034] have already been one of the most important infrastructures of the Internet nowadays. For example, DNS is an indispensable system of the Internet used for translating the "human-friendly" host names of computers on a TCP/IP network into their corresponding "machine-friendly" IP addresses. In general, DNS also stores other types of information, such as the list of mail servers that accept email for a given Internet domain. By providing a worldwide, distributed name service, DNS is an essential component of the functionality of the Internet.

Similarly, global name services will also be one of essential and key elements in constrained networks, which can be used for translating the "thing-friendly" names of constrained devices or RFID tags on a lightweight TCP/IP network into their corresponding "machine-friendly" addresses or other related information of another constrained network. The applications or devices on a constrained network can easily communicate with other devices on the same or any other constrained network with the name of the constrained device by a global name service, without considering whether the address of the targeted constrained device on a constrained network has been changed

or not.

To fulfill the aforementioned objective, a lightweight global name service based on the LWIP protocol needs to be researched. The efficiency of this kind of service is supposed to be the most important issue to be studied in future.

6.4. Device configuration

There are several type of devices for several purposes, and several applications running on the devices in the scenarios. There are some devices which has a monitor and a keyboard to monitor and to control the networks for administrating. But, most devices like a sensor, or a actuator do not have enough space to have such interface. It means there is no configuration interface of the device. The constrained device sometimes have to be configuared via an flash memory, or another interface like USB. However, it is difficult to set up handreds of devices in the networks. Furthermore, some types of devices like a temperature sesor do not have a space for it. In this scenario, it is recommended to configure the devices via the network. And, special messaging and special format are not scalable. We should have to consider the method to configure the device via a similar way.

7. Security Considerations

TBD.

8. Acknowledgement

TBD.

9. IANA Considerations

This document does not require any IANA actions.

10. Normative References

[I-D.moskowitz-hip-rg-dex]
Moskowitz, R., "HIP Diet EXchange (DEX)", draft-moskowitz-hip-rg-dex-02 (work in progress), July 2010.

[RFC0813] Clark, D., "Window and Acknowledgement Strategy in TCP", RFC 813, July 1982.

[RFC0817] Clark, D., "Modularity and efficiency in protocol implementation", RFC 817, July 1982.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[tinyos] "Tiny Operation System on Mica Series Sensors", <www.tinyos.net>.

[uip] "Full TCP/IP for 8-bit Architectures", <http://www.sics.se/~adam/uip/index.php/Main_Page>.

[usd] Canfeng Chen, "uSD: an SD-based Mobile Gateway to Wireless Sensor Network".

Authors' Addresses

Hui Deng
China Mobile
Unit2, 28 Xuanwumenxi Ave,Xuanwu District
Beijing 100053
China

Email: denghui@chinamobile.com

Shoichi Sakane
Cisco
2-1-1 Nishi-Shinjuku, Shinjuku-ku
Tokyo 163-0409
Japan

Email: ssakane@cisco.com

Wassim Michel Haddad
Ericsson
300 Holger Dr
San Jose, CA 95134
US

Phone: +1 646 256 2030
Email: Wassim.Haddad@ericsson.com

Ning Kong
CNNIC
4 South 4th Street,Zhongguancun,Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 3147
Email: nkong@cnnic.cn