Network Working Group
Internet-Draft
Intended status: Experimental
Expires: April 17, 2016

M. Stenberg

October 15, 2015

Simple Home Status Protocol
draft-stenberg-shsp-01

Abstract

In reasonable home networks there are no cloud services; just a set of local nodes sharing state, and possibly invoking operations either automatically or on behalf of the home user. This document describes a key-value database that can be used to do both; published key=values represent the current state of the system, and by setting a key published by some other node to a different value a node can indicate the desire for changing of state.

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 17, 2016.

Table of Contents

1. Introduction

In spirit of House Arkko Toaster/Laundry/et al, House Stenberg also has needs, although they are mostly related to various sensors, control of lights and various appliances. This leads to a a number of devices that need to discover each other, and share state and code (ideally) without central authorities, in fault tolerant fashion. And obviously it is also nice to have a log of what has occurred, and maintain software state in a distributed fashion. There are also some security needs; control of appliances can be a serious hazard if security is not addressed appropriately. SHSP is an attempt to produce a solution to address these needs.

This document describes yet another DNCP-based protocol, which uses HNCP-style transport, yet intentionally is incompatible with it so that TLVs of SHSP and HNCP can be handled using same transport channel and de/encoders, but individual implementations of the different protocols can ignore each other unless they support both protocols.

TBD: The long form of the acronym is actually misleading, as it is more about independent agents rampaging in home network, sharing state as they go. Figure a better name?

2. Requirements language

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 RFC 2119 [RFC2119].

3. DNCP Profile

Hard work of SHSP is handled by DNCP [I-D.ietf-homenet-dncp]. DNCP profile of HNCP [I-D.ietf-homenet-hncp] is used as is. Arguably TCP-based solution would be better, but compatibility with existing implementations of HNCP to transfer state, and eventual port- and multicast address reservations of HNCP make it more attractive. Despite the reuse of the transport and DNCP core logic, NONE of the HNCP TLVs are part of SHSP.

SHSP nodes MAY choose to use the DTLS method(s) specified in the DNCP document, but in practice if authentication is desirable, something more lightweight using simple PSKs + sha256 authentication may be used. To prevent providing a bad example, a concrete example is omitted.

4. Overview

SHSP provides a distributed key-value store, with each node having consistent view of the whole database. Essentially, the database provides view of ((int) node identifier, (string) key, (json-encodable) value, (UTC second)last changed at) pairs for any participant interested in viewing it.

If we assume that keys in and of themselves are unique, then we can assume that someone else than the node responsible for particular resource belonging to that key publishing a key=value for it desires a state change, e.g. turning on a light.

5. Device Data

SHSP devices typically provide either boolean input/output, or some numeric output value for other nodes to use. Whatever can be transmitted inside JSON (particular key = particular value) is fine, but for example, in House Stenberg there are currently:

- Light sensor: numeric value (in lux) .kh.light_sensor.(name)/value=(int).
- Philips Hue lights: boolean with key .kh.hue.(name)/on. (TBD: this could be also active color value)

- Motion sensor: boolean with key .kh.motion_sensor.(name)/on.

- Wifi device presence detection: boolean key with .kh.wifi.(name)/on.

- Process activity monitor: boolean key with .kh.process.(name)/on.

- (Computer) user activity monitor: boolean key with .kh.user_active.(name)/on.

Very cut down example of what the state actually looks like in the whole network of devices, that is determined by looking at the whole DNCP network state, not just that of the local node, looks like this at the time of the writing of the document:

```
mstenber@poro ~>khtool -v
.kh.hue.Entry/on=False (-38)
.kh.light_sensor.corridor/value=86 (-37)
.kh.motion_sensor.corridor/on=False (-49)
.kh.process.kodi/on=False (-67807)
.kh.user_active.poro/on=True (-41)
.kh.wifi.iphone6/on=True (-5718)
...
```

The numbers in parentheses identify second delta to current time when the state last changed. For example, apparently Kodi process was last running on the machine it is monitored on ~20 hours ago, and the light called Entry was turned off 38 seconds ago.

6. Type-Length-Value Objects

SHSP requires only one TLV for its own use:

6.1. SHSP Key-Value State TLV

TBD: ASCII art

- T=789 (TBD)

- V= JSON-encoded string, containing a dictionary with:

  - ts: (timestamp) when was the value most recently changed; number, expressed in seconds, UTC since UNIX epoch

  - k: which key the value applies for

  - v: the value of the key (any acceptable Json)

For example, the value of the TLV describing Kodi state noted in the above example looks as follows: {"k": ".kh.process.kodi/on", "v": false, "ts": 1444598358.729439}

7. Security Considerations

N/A

8. IANA Considerations

This document has no actions for IANA as it is informational for fun document, not really aimed for standards track.

9. 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, <http://www.rfc-editor.org/info/rfc2119>.

[I-D.ietf-homenet-dncp] Stenberg, M. and S. Barth, "Distributed Node Consensus Protocol", draft-ietf-homenet-dncp-09 (work in progress), August 2015.

[I-D.ietf-homenet-hncp] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", draft-ietf-homenet-hncp-09 (work in progress), August 2015.

Appendix A. Changelog

draft-stenberg-shsp-01:

- Got rid of the Python agent section - while it is planned at some point, not implemented yet and I am mostly interested in documenting the current production scheme.
- Added concrete examples of sensors and current system output.

Appendix B. Draft source

As usual, this draft is available at https://github.com/fingon/ietf-drafts/ in source format (with nice Makefile too). Feel free to send comments and/or pull requests if and when you have changes to it!

Appendix C. Acknowledgements

None yet, want to be the first?

Author's Address

Markus Stenberg
Helsinki 00930
Finland

Email: markus.stenberg@iki.fi