PAWS Database Discovery

draft-wei-paws-database-discovery-00

PAWS WG

IETF86

Xinpeng

Zhu Lei

1. Motivation

- The pre-configuration and provisioning are valid approaches for WS master devices to get DB addresses.

- Device management and other provisional approaches are to retrieve DB address in operator networks.

- This proposal is a light use of LoST (RFC5222) for WSDB discovery, with advanced services of LoST

  – In case, WSD is failure to access pre-configured or cached DB server.

2. Services LoST provides to DB discovery

- LoST (RFC5222) architecture and services
  - Mapping locations to PSAP URLs
  - Architecture considerations: seekers, resolvers, trees, and forest guides
    - Find LoST service by 'DHCP extension to LoST' (RFC5223)
    - WS master device behaves as seekers, receives DB address from resolvers.
    - Recursion and Iteration help resolve DB address list to location
  - Features useful to PAWS DB discovery:
    - Inputs: location profiles, output: service list, DB server(s)
    - Address validation
    - Service boundaries

3.Specification

```
+--------------+
|  LoST        +
+--------------+
|  HTTPS       +
+--------------+
|  TCP         +
+--------------+
|  IP          +
+--------------+
```
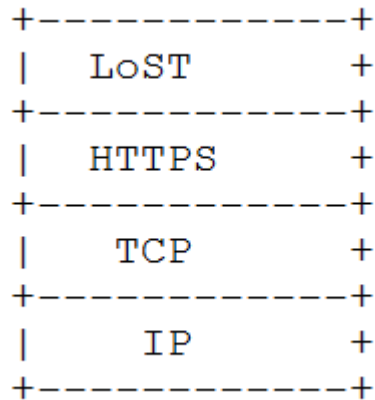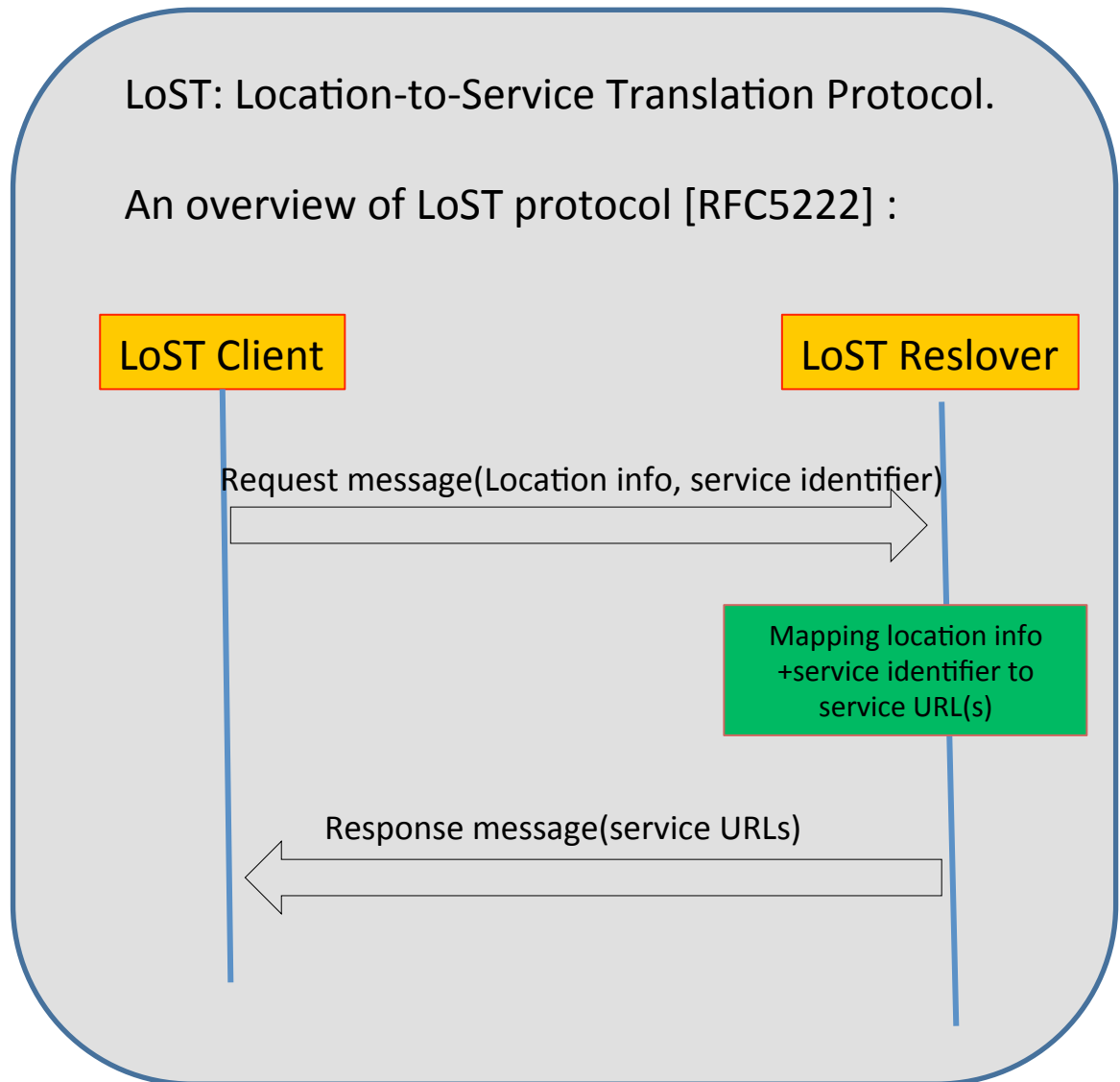
Figure : Protocol stack for discovery mechanism

Location info: LoST supports GPS and postal address
Service Id.: needs IANA registration.
Extensions to LoST protocol to address issues ?
For example, if we need region while sending requests

LoST: Location-to-Service Translation Protocol.

An overview of LoST protocol [RFC5222] :

| LoST Client | | LoST Reslover |

Request message(Location info, service identifier)

Mapping location info +service identifier to service URL(s)

Response message(service URLs)

4. Issues to be clarified further

- Paws uses JSON, LoST encodes as xml: preference option is using HTTP boundary mark, otherwise, goes to specify LoST JSON encoding.

- Conveying of regulatory domain

  The regulatory domain information needs to be conveyed to the WSD for later use.

  Pro: shorten the list of WSDB service list. Quickly resolve WSDB DS and DB addresses.

- Location information of longitude and latitude as inputs to LoST server may add region info, in case a master device request a serving LoST server.

4a. Issues to be clarified further

- WSDB DS process is to ensure the ability to find serving DB server for master devices.

- By applying LoST service to WSDB discovery, how LoST and WSDB discovery service deployed? Possible owners of WSDB service records: ISP sor DB service providers.

- Recursion and iteration can resolve two cases.

4b. Issues to be clarified further

- Still need paws protocol to Mutual Authentication
  - For case of a number of DB URLs received.
  - Keep business relationship of DB service provider and DB master devices.

- If WSDB discovery needs regulatory domain information?
  - Pro: prioritize DB address by regulatory domain and service info. Choose proper DB to connect.

Questions to follow up or confirm in mail list

- 1） LoST service discovery
- 2) WSDB discovery service, location/WSDB service mapping provided by ISP, or provided by WSDB service providers
- 3) WSDB DS responses multiple DB addresses, how business relationship works when receiving address list
- 4) WSDB DS could response DB address by receiving GPS information and/or postal address.
- 5) Question: If WSDB SD request would include region information with GPS information?
- 6) LoST relies on U-NAPTR, and we all know how difficult is to get DNS admins add new records to their zones.
- 7) The WG seems to prefer JSON encoding to XML, to be able to support lightweight devices, LoST uses XML.

Thanks!!!

Backup a.

- Discovery Request procedure

An example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <findService
    xmlns="urn:ietf:params:xml:ns:lost1"
    xmlns:p2="http://www.opengis.net/gml"
    serviceBoundary="value"
    recursive="true">
    <location id="6020688f1ce1896d" profile="geodetic-2d">
      <p2:Point id="point1" srsName="urn:ogc:def:crs:EPSG::4326">
        <p2:pos>37.775 -122.422</p2:pos>
      </p2:Point>
    </location>
    <service>urn:service:paws.discovery</service>
  </findService>
```

location info

service identifier

Backup b.

- Discovery Response procedure

An example:

regulatory body

```
<displayName xml:lang="en">
    Federal Communications Commission
</displayName>
<service>urn:service:paws.discovery</service>
<serviceBoundary profile="geodetic-2d">
    <p2:Polygon srsName="urn:ogc:def::crs:EPSG::4326">
        <p2:exterior>
            <p2:LinearRing>
                <p2:pos>37.775 -122.4194</p2:pos>
                <p2:pos>37.555 -122.4194</p2:pos>
                <p2:pos>37.555 -122.4264</p2:pos>
                <p2:pos>37.775 -122.4264</p2:pos>
                <p2:pos>37.775 -122.4194</p2:pos>
            </p2:LinearRing>
        </p2:exterior>
    </p2:Polygon>
</serviceBoundary>
<uri>database1.example1.com</uri>
<uri>database2.example2.com</uri>
```

service boundary info

service URLs

Backup c.

- ## Service Identifier

  A new service identifier for PAWS database discovery needs to be defined.

  According to RFC5031 [RFC5031], a top-level service and a sub-service
  are defined here.

```
+----------------+----------------------------------------+
|    Service     |               Description              |
+----------------+----------------------------------------+
|      paws      |        top-level service of PAWS       |
| paws.discovery | the PAWS database discovery service    |
+----------------+----------------------------------------+
```