DAPNET 2.0 Concept and Interface Definition

Ralf Wilke, DH3WR
Thomas Gatzweiler, DL2IC
Phillip Thiel, DL6PT

July 4, 2018

Abstract

This is the concept and interface description of the version 2 of the DAPNET. It's purpose in comparison to the first version released is a more robust clustering and network interaction solution to cope with the special requirements of IP connections over HAMNET which means that all network connections have to be considered with a WAN character resulting in unreliable network connectivity. In terms of consistence of the database, "eventually consistence" is considered to be the most reachable. There are "always right" database nodes inside the so called HAMCLOUD. In case of database conflicts, the version inside the HAMCLOUD cluster is always to be considered right.

1 Introduction

write some history

2 Concept presentation

2.1 Core functionality

2.2 HAMCLOUD presentation

The HAMCLOUD is a virtual server combination to server central services on the HAMNET and provide short hop connectivity to deployed service on HAMNET towards the Internet. There are three data centers at Essen, Nürnberg and Aachen, which have high bandwidth interlinks over the DFN. There are address spaces for uni- and anycast services. How this concept is deployed is still tbd. More information is here https://www.swiss-artg.ch/fileadmin/Dokumente/HAMNET/HamCloud_-_Angebotene_Dienste_in_der_HamCloud.pdf and here http://hamnetdb.net/?m=as&q=hamcloud.

Define if uni- or anycast entry points will exist

3 Interface definition

3.1 Transmitter sign-in, configuration and keep-alive

If a transmitter wants to connect to DAPNET, the first step is to sign-in and show it's presence via a REST interface. Tis interface is also used for transmitter configuration like enabled timeslots and keep-alive polling.

3.1.1 Authentication of all HTTP-Requests in this context

All HTTP-requests issued from a transmitter have to send a valid HTTP authentication, which is checked against the CouchDB. It consists of the transmitter name and its AuthKey.

3.1.2 Initial contact

POST /transmitter/bootstrap

```
{
  "callsign": "db0avr",
  "auth_key": "<secret>",
  "software": {
    "name": "UniPager",
    "version": "1.0.2"
  }
}
```

3.1.3 Answers to the bootstrap REST call

200 OK

```
{
  "timeslots": [true, true, false, true, ...],
  "nodes": [
    {
      "host": "node1.ampr.org",
      "port": 4000,
      "reachable": true,
      "last_seen": "2018-07-03T07:43:52.783611Z",
      "response_time": 42
    }
  ]
}
```

423 Locked

```
{
  "error": "Transmitter temporarily disabled by config."
}
```

423 Locked

```
{
  "error": "Transmitter software type not allowed due to serious bug."
}
```

3.1.4 Transmitter Heartbeat

POST /transmitter/heartbeat

```
{
  "callsign": "db0avr",
  "auth_key": "<secret>",
  "ntp_synced": true
}
```

3.1.5 Answers to the heartbeat REST call

200 OK

```
{
  "status": "ok"
}
```

If network wants to assign new timeslots without disconnecting (for dynamic timeslots)

200 OK

```
{
  "status": "ok",
  "timeslots": [true, true, false, ...],
  "valid_from": "2018-07-03T08:00:52.786458Z"
}
```

If network wants to initiate handover to other node

`503 Service unavailable`

```
{
  "error": "Node not available, switch to other node."
}
```

4 Visualization and management Website Connections

There will a responsive website will be the main user interface. Dynamic data to be displayed is transfered by two methods:

4.1 Polling based initial data dump

When the website is called, the DAPNET REST API is use to gather the required data according to content just displayed. The API is defined in section .

write API docu

4.2 Websocket based updates

Once the actual status is known via the REST call, dynamic updates and telemetry data is send to the user's browser via websocket. Authentication is done by a authentication message by the browser right after connection. This procedure is the same as used in unipager.

4.2.1 Check authentication status

DL2IC: Könntest du das ergänzen?

4.2.2 List of update messages

The updates can be send combined in one message of each message separately. In any case, the *transmittername* has to be sent to identify about what transmitter the data is.

`Complete Telemetry, sent every xxx Seconds`

`Status Update`

```
{
    "name": "db0acb",
    "transmitterupdate" : {
        "status" : {
            "OnAir" : true
        }
    }
}
```

`Transmitter configuration Update`

```
{
    "name": "db0acb",
        "transmitterupdate" : {
                "status" : {
                        "ConfiguredIP": "123.4.3.2",
            "timeslots" : [true, false,...,     false],

                        "SoftwareType" : "Unipager",
            "SoftwareVersion" : "v1.2.3",
or
                        "SoftwareType" : "MMDVM",
                        "SoftwareVersion" : "20180504",
or
                        "SoftwareType" : "DAPNET-Proxy",
                        "SoftwareVersion" : "v2.3.4",
                        "CPUHardwareType" : "Raspberry Pi 3B+"
                        "RFHardware" : {
                                "C9000" : {
                                        "UnipagerPowered" : true,
                                        "ArduinoPADummy" : true,
                                        "ArduinoPADummySettinginWatts" : 123,
                                        "ArduinoPADummyPort" : "/devttyUSB0"
or
                                        "RPC-CardPowered" : false,
                                        "RPC-Version" : "XOS/2.23pre",
                },
                "Raspager" : {
                    "RaspagerMod" : 13,
                    "RaspagerPower" : 63,
                    "ExternalPowerAmplifier": false,
                    "RaspagerRFVersion" : "V2"
                },
                "Audio" : {
                    "TXModel" : ["GM1200", "T7F", "GM340", "FREITEXT"],
                    "AudioLevelUnipager" : 83,
                    "TxDelayinMilliseconds" : 3
                }
                "RFM69" : {
                        "Port" : "/dev/ttyUSB0"
                },
                "MMDVM␣DualHS..." : {
                    "DAPNETExclusive" : true
                }
        }
    }
}
```

DAPNET-Proxy Update

```
{
    "name": "db0acb",
    "transmitterupdate" : {
        "status" : {
            "AX25" : {
                "ConnectionStatus" : "connected",
                "ConnectionStatus" : "connecting",
                "ConnectionStatus" : "disconnected"
            }
        }
    }
}
```

Telemetry Update Sent every minute complete, if there are changes, just a subset is sent.

```
{
    "name": "db0acb",
    "Telemetry" : {
        "ConnectionStatus": {
            "Connected" : true,
            "ConnectedtoNodeName" : "db0xyz"
            "ConnectedtoNodeIP" : "1.2.3.4"
            "ConnectedtoNodePort" : 1234
            "ConnectedSince" : "<timestamp-format>",
                "NTPSynced" : true,
            "NTPOffestMilliseconds" : 124
            "NTPServerUsedIP" : ["134.130.4.1", "12.2.3.2"]
            }
        },
    "QueueStatus" : {
        "Telemetry" : {
            "QueuedMessages" : {
                "Total" : 1234,
                "Prio1": 1234,
                "Prio2": 1234,
                "Prio3": 1234,
                "Prio4": 1234,
                "Prio5": 1234,
                "Prio10": 1234
            },
        "PrefinedTemperatures" : {
            "Unit" : "C",
or
            "Unit" : "F",
or
            "Unit" : "K",

            "AirInlet" : 12.2,
            "AirOutlet" : 14.2,
            "Transmitter" : 42.2,
            "PowerAmplifier" : 45.2,
            "CPU" : 93.2,
            "PowerSupply" : 32.4
        },
        "CustomTemperatures" : {
            "Unit" : "C",
or
            "Unit" : "F",
or
            "Unit" : "K",
            [
                {"Value" : 12.2, "Description" : "Aircon Inlet"},
                {"Value" : 16.2, "Description" : "Aircon Outlet"},
                {"Value" : 12.3, "Description" : "Fridge Next to Programmer"}
            ],
        },
        "PowerSupply" : {
            "OnBattery": false,
            "OnEmergencyPower": false,
            "DCInputVoltage" : 12.4,
            "DCInputCurrent" : 3.23
        },
        "RFoutput" : {
            "OutputPowerForwardinWatts": 12.2,
            "OutputPowerReturninWatts" : 12.2,
            "OutputVSWR" : 1.2
        }
    }
}
```

5 CouchDB Document Structure

5.1 Users

```
{
    "users": {
        "name": "dl1abc",
        "password": "some hash",
        "email": "user@example.com",
        "admin": true,
        "enabled":true,
        "created_on":<DATETIME>,
        "last_change_by":"dh3wr",
        "email_valid":true
        "avatar_picture": <couchdb attachment??>
}
```

5.2 Nodes

```
{
    "nodes" : {
        "name" : "db0abc",
        "status" : "OFFLINE",
or
        "status" : "ONLINE",
or
        "status" : "ERROR",
        "last_update" : DATETIME,
        "version" : "1.2.3",
        "ip_address" : "1.2.3.4",
        "latitude" : 34.123456,
        "longitude" : -23.123456,
        "hamcloudnode" : true,
        "owners" : ["dl1abc","dh3wr","dl2ic"],
        "avatar_picture": <couchdb attachment??>
}
```

5.3 Transmitters

```
{
    "transmitters" : {
        "name" : "db0abc",
        "auth_keys" : "hdjaskhdlj",
        "enabled" : true,

                "status" : "UNKNOWN",
or
                "status" : "OFFLINE",
or
                "status" : "ONLINE",

                "usage_type" : "PERSONAL",
or
                "usage_type" : "WIDERANGE",

        "aprs_enabled" : true,
        "last_update" : "<DATETIME>",
                "last_connect" : "<DATETIME>",
                "connected_since" : "<DATETIME>",
        "ip_address" : "1.2.3.4",
        "device_type" : "Unipager",
        "device_version" : "1.3.2",
        "latitude" : 23.123456,
        "longitude" : -31.123456
        "rf_power_watt": 12.3,
        "cable_loss_db" : 4.2,
        "antenna_gain_dbi" : 2.34,
                "antenna_agl" : 23.4,

                "antenna_type" : "OMNI",
or
                "antenna_type" : "DIRECTIONAL",

                "antenna_direction" : 123.2,
        "owners" : ["dl1abc","dh3wr","dl2ic"],
        "member_in_transmittergroups" : ["dl-hh", "dl-all"],
        "frequency" : 439.9875,
        "emergency_power_available" : false,
        "infinite_emergency_power" : false,
                "emergency_power_duration_hours" : 23.0
        "antenna_pattern" : <couchDB attachment>,
        "avatar_picture" : <couchDB attachment>
    }
}
```

5.4 Transmitter Groups

Transmitter Groups are logical associations of one or more transmitters. This structure just defines the meta data of a transmitter group, the membership is part of the transmitter storage itself.

```
{
    "transmitter_groups" : {
        "name" : "dl-hh",
        "description" : "Deutschland-Hamburg",
        "owners" : ["dl1abc","dh3wr","dl2ic"],
    }
}
```

5.5 Subscribers

check if [] is valid JSON

```
{
    "subscriber" : {
        "name" : "dl1abc",
        "description" : "Peter",
        "pagers" : [
                {
                "ric" : {123456, "A"}
till
                        "ric" : {123456, "D"},

                        "uuid" : "0023-1233-aefe-1234-3423-9812",
                "name" : "Peters Alphapoc",

                        "type" : "UNKNOWN"
or
                        "type" : "Skyper",
or
                        "type" : "AlphaPoc"
or
                        "type" : "QUIX",
or
                        "type" : "Swissphone",
or
                        "type" : "SCALL_XT"
or
                        "type" : "Birdy"

                        "is_enabled" : true
        }, ...
        ],
            "owner" : ["dh3wr", "dl1abc"]
    }
}
```

5.6 Subscriber Groups

```
{
    "subscriber_group" : {
        "name" : "ov-G01",
        "description" : "Ortverband Aachen",
        "member_subscribers" : ["dl1abc", "dh3wr"],
        "owner" : ["dh3wr", "dl1abc"],
    }
}
```

5.7 Rubrics List

```
{
    "rubrics" : {
        "uuid" : "<UUID>"
        "number" : 14,
        "description" : "Wetter DL-HH",
        "label" : "WX DL-HH",
        "transmitter_groups" : ["dl-hh","dl-ns"]
        "cyclic_transmit_enabled" : true
        "cyclic_transmit_interval_minutes": 123,
        "owner" : ["dh3wr", "dl1abc"]
    }
}
```

5.8 Rubric's content

<UUID> of rubric (as defined in 5.7)

```
{
    "rubric_content" : {
```

```
                "uuid" : "<UUID>",
                ["content_message1",..,"content_message10"],
    }
}
```