Greedy Perimeter Stateless Routing (GPSR)

README

(July 6 2003)

Young Jin Kim (Youngjki@usc.edu)
Ramesh Govidan (Ramesh@usc.edu)

Embedded Network Lab
Department of Computer Science
University of Southern California
1. Introduction
This document describes targets where GPSR daemon is run-able, how to build GPSR daemon and set compiling option, how to run GPSR daemon and set running parameters, and Limitations.

2. Targets
- PCs/WSs running Red-hat Linux 8.0/9.0
- PC-104 running Linux + Real mote(Mica)
- HP IPAQ running familiar Linux -http://www.handhelds.org

3. Getting started
First of all, you can get a software package from http://enl.usc.edu/software/index.html. It consists of GPSR daemon, Library for API, and a sample application.

The directory structure of s/w package is as follows:
```
S/ W ---- gpsr          # files for GPSR daemon
    |--- gpsr_oam     # files for maintaining daemon(change of location, beacon period., etc)-Optional
    |--- gpsr_api      # files for application program interface
    |--- apps -- sample # files for a sample application
```

![Figure 1. Overall Software Structure](image)

4. Building GPSR daemon
Before building GPSR daemon, you should check your target and compiling environments.
- In case of PC, GNU compiler (gcc) is needed for building GPSR.
- In case of PC104, pc104-linux-gcc is needed.
- In case of HP IPAQ, there is two ways for compiling software: 1) direct-compiling on HP IPAQ where gcc and make are setup, 2) cross-compiling on your HOST by using arm-linux-toolchain (http://handhelds.org/download/toolchain).

After compiling environments is identified, update Makefile to use a proper compiler. Also, you should update Makefile to determine the size of communication buffer between application and gpsr daemon, and the size of packet payload among neighbor nodes.
```
-DMAX_BUFF_SIZE=300 -DMAX_PAYLOAD_SIZE=150
```

Finally, you stand in front of GPSR world: type `cd gpsr` and `make install`.
If compiling is successful, you can find running-object `gpsr_d` file that cares of GPSR functionalities. To make an application using GPSR daemon, type `cd gpsr_api` and `make install`.
If compiling is successful, you can find library `libgpsr.a` that provides API.
After then, go to your application directory. Before compiling your application, you should confirm that install line in Makefile has `$('$(GPSR_API_DIR)/libgpsr.a')`.

For example, install:

```
(C++) -o index $(OBJS) $(GPSR_API_DIR)/libgpsr.a $(LD_FLAGS) $(LD_LIBS)
```

5. Running GPSR daemon

To let you know running options, type `gpsr_d -h`. It will show you the followings.

Usage: gpsr [OPTION...]
-d, --debug # Runs in debug mode
-G, --GG # Set gpsr_mode to GG [Default mode]
-m, --oam_port # Set gpsr-maintenance's port number [Default port is 5002]
-p, --daemon_port # Set gpsr-daemon's port number [Default port is 5001]
-R, --RNG # Set gpsr_mode to RNG
-t, --beacon_time # Set inter beacon timer's value [Default is 40 seconds]
-x, --x-position # Set my x-position [Mandatory]
-y, --y-position # Set my y-position [Mandatory]
-h, --help # Display this help and exit

To run gpsr_d, you should type the followings:

```
gpsr_d -x 1000 -y 1000 -t 35 # position of your node is (1000,1000) and beacon timer is 35 seconds
gpsr_d -x 1000 -y 1000 -t 35 -d # if you want to look at the running status of gpsr_d
```

6. Limitations

- No interface between GPSR daemon and Localization function.
- If you use GPSR_oam to change location, it will emulate localization interface.
- No support of Mutual GG or RNG
- The limited size of coordinate plane:
  - In case of PC104, GPSR daemon supports up to 256x256 plane.
- Up to now, software has been experimented on Ethernet and AP mode 802.11b environment. In other words, we can't confirm running on Ad-hoc mode 802.11b. As soon as possible, we will do it.

7. References