
# Executive Overview

TITLE: D3.2.2-Design and implementation of CARLINK wireless ad-hoc applications: Finding and Sharing Files

SUMMARY: This report presents FSF (Finding and Sharing Files) as an application for searching and sharing files in VANETs (Vehicular Ad-hoc NETworks).

GOALS:

1. Introducing the motivations for file sharing in VANETs.
2. Enumerating FSF features.
3. Describing the real settings for testing FSF.
4. Summarizing the most significant results of the real experiment.

CONCLUSIONS:

1. After analyzing the obtained results of the real experiments we conclude that FSF is ready to work in real VANETs.

# D3.2.2-Design and implementation of CARLINK wireless ad-hoc applications: Finding and Sharing Files

CARLINK::UMA

November, 20th 2007

## 1 Introduction

File sharing is always an interesting service in every communication network because of the wide range of applications that could make use of it, in particular it is an useful service for CARLINK primary applications: urban transport traffic management, local weather data, and information broadcasting/sharing. However, achieving successful transfers in VANETs is a challenging task due to several factors: nodes mobility, appearing obstacles between nodes, and reflection problems inherent to urban environmnets.

In this report we presents an application for sharing files in VANETs, called FSF (Finding and Sharing Files). The remainder of this report is organized as follows: Section 2 describes the FSF features and explains how it works. Section 3 specifies the hardware and the software for running FSF. Finally, Section 4 presents the more significant results obtained during the real tests of FSF and the conclusions about it.

## 2 Featuring FSF

The idea behind FSF is that every node in the VANET has a special folder, called **shared folder**, where placing the files to be shared with all the others nodes. FSF provides two main functionalities: **searching** files in every shared folder of the whole network and **downloading** what they contain.


a) Tab for searching and downloading files

b) Tab for managing the shared folder

Figure 1: FSF Graphical User Interface

## 2.1 File searching

The process of searching files begins when the user enters one key-word (see Figure 1a) identifying the name of the file which is being looked for. This key-word is inserted in a search message after pushing the **search** button. Then, the search message is sent to all the other nodes in the VANET.

In order to reach all the nodes in the network, it is necesary to use *flooding techniques*. The behaviour of the algorithm is not so difficult. One initial node broadcasts the search message among all the adjacent nodes and every node that receives this message repeats this action. Each message has an unique identifier and every time a node receives one message, its identification is stored in a table. This table is checked whenever a node receives a new message. If the identification of the actual message is found in the table, the message has been sent before and will not be sent again.

Moreover, the key word inside the search message is used when it is received to pick up the names of the files in the shared folder matching this key word. This names are packaged in a new message that will be sent back to the node who started the searching process. If the key word does not match with any file name, the node has nothing to do.

## 2.2 File downloading

The results of the file searching are shown in a table (see Figure 1a). This table offers information about the complete name of each file in the network matching the key word of the search process, the IP address of the devices where they are placed into and the distance of these devices (number of hops). Thus, the user is able to apply his own selection criteria (reliability, proximity, etc.) in each particular situation for downloading the file from the desired device.

Once the user decides the file to download, he/she only needs to double click on the row of the table where it appears and push the **download** button. Then, FSF starts the downloading process from the selected source. The transfer of the file is carried out by the VDTP procotol. This is a special purpose file transfer protocol designed for VANETs which provides a reliable service by using inter-vehicular communications. Detailed information about its behaviour and performance can be found in [1]. Basically, VDTP has two phases: the first one is for querying the file size and negotiating the chunk size for splitting the file, and the second one is for data and confirmation packages interchange until the file has been completely downloaded. In this second phase, the transference is done chunk by chunk. The chunk number i+1 is not requested until the confirmation for the chunk number i has arrived.

It is possible to configure the size of the chunk to be used every time a file is being downloaded. This feature makes FSF adaptable to different situations. For example, if the users who wants to interchange files are adjacents and the network usage is scarce, they should select large size for the chunks (30KB - 60KB) in order to achieve high transference rates. However, if the users are not adjacents or the networks traffic is intensive, they should select small size for the chunks (less than 30KB) for ensuring that the transfer is successfully done.

## 2.3 Managing the shared folder

The FSF graphical user interface (see Figure 1b) allows to modify the shared folder content by adding new files or deleting existing ones. It also shows information about the shared files, such as: the complete file name and the file size in bytes.

The user is able to share new files by pushing the **add** button or by using the windows manager of the operating system, i.e., manually. When the add button is pushed, FSF opens a small window for asking the location of the new file to be shared. Once the file is selected, the file is copied from its actual location to the shared folder. If the user selects to do it manually, he must push the **refresh view** button afterwards; this is needed to update the status information about the shared folder. Anyway, the result is a new file which can be downloaded by all the others nodes in the network.

In the same way, there are two options for deleting existing files from the shared folder: pushing **delete** button or using the operating system windows manager. Once again, it is necesary to remember that the refresh view button has to be pushed after deleting the file manually. If not, FSF does not offer coherent information about the shared folder content. Concretely, when any user is looking for the deleted file, FSF answers as it still exists. However, when the download request arrives, an error occurs and no file is transferred since the file does not exist.

# 3 Execution Platform

FSF has been implemented with JANE [2], a Java-based middleware which is intended to assist ad-hoc network researchers in application and protocol design. FSF has been executed in laptops running the Java Virtual Machine of *Sun Microsystems*. These laptops were equipped with *ORiNOCO IEEE 802.11b/g PC Card GOLD*[1] connected to a range extender antenna, with 7dB gain, in order to create an effective VANET.

# 4 Results and conclusions

The system was tested several times during March 2007. Concretely, the crucial tests were performed on the 13th and 22th March, using two cars across the parking placed at the *E.T.S. Ingeniería Informática*[2] in the *University of Málaga*.

The experiment was composed of several tests. Each test consisted of transferring a file between two cars moving along 500m at 30 km/h. We used two different trajectories for the cars and two different files for testing the application. The movement of the vehicles is described in the scenario A (see Figure 2) and scenario B (see Figure 3). The files used had 1 MB size (representing documents) and 10 MB size (representing multimedia files).

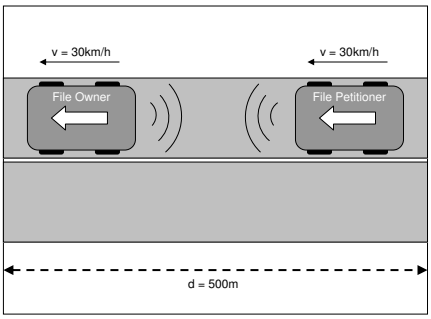


Figure 2: Scenario A

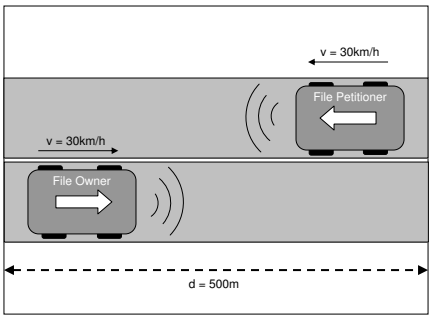


Figure 3: Scenario B

All the obtained results are included in [1]. The best transferring rates reached 626KB/s on average, during the transfer of 1 MB file in scenario A, while the worst transferring rates reached 371KB/s on average, during the transferring of the same file in scenario B.

Note that it is a first attempt to obtain real results from file transferring in real VANETs. Therefore, the scenarios for testing the system were simple. However, the obtained results showed that it is possible to move forward more complex tests. At the moment, we conclude that FSF is ready to work in simple VANETs what could be interesting for the near future inside the CARLINK project.

[1] Proxim Wireless Networks - `http://www.proxim.com`
[2] School of Computer Science Engineering - `http://www.informatica.uma.es`

# References

[1] CARLINK::UMA. D2006/10 - VDTP: A File Transfer Protocol for Vehicular Ad-hoc Networks. Technical report, University of Malaga, Spain, 2006.

[2] CARLINK::UMA. D2006/7 - JANE: A Tool for Implementing Applications in Real Ad-hoc Networks. Technical report, University of Malaga, Spain, 2006.