Executive Overview

TITLE: VDTP: A File Transfer Protocol for Vehicular Ad hoc Networks

SUMMARY: This report presents a suitable protocol for file transferring in highly dynamic environments such as vehicular ad hoc networks.

GOALS:

1. Establishing the main requeriments for file transfer in vehicular ad hoc networks.
2. Featuring the protocol behavior.
3. Showing the protocol performance evaluation in real settings.

CONCLUSIONS:

1. The obtained results during the real performance evaluation allow us to conclude that VDTP is suitable for CARLINK proposals.

VDTP: A File Transfer Protocol for Vehicular Ad hoc Networks

Sebastián Luna

May 26, 2008

1 Introduction

Vehicular Ad hoc Networks (VANETs) are self-configuring networks, established by using cars (mobile nodes) with wireless interfaces, communicating with each other in a peer-to-peer fashion. VANETs are a hostile communication medium due to several factors [6]:

- Node velocity, which implies frequent changes in network topology.
- Appearing obstacles between transmitters and receivers, leading to broken connections.
- Reflection problems, caused by urban environments.

File transferring is always an interesting service in every communication network because of the wide range of applications that could make use of it, in particular, for VANETs and CARLINK primary applications: urban transport traffic management, local weather data, and information broadcasting/sharing. However, achieving successful transfers in VANETs is a challenging task owing to the adverse factors mentioned above. Therefore, it is necesary to take into consideration the following requeriments when building a reliable file tranfer protocol for VANETs:

- Connection-oriented communication protocols, such as TCP, are designed for static networks. Therefore, the use of these kinds of protocols in a highly dynamic environment, such as VANETs, would induce significant control overhead for connection maintenance, resulting in increasing network traffic and low effectiveness.
- Due to the changing network topology, the sender location may change during a file transfer. Therefore, it is essential to keep the complete control over the transfer on the receiver side.
- In the absence of fixed infrastructures, nodes have to cooperate to provide routing services, relying on each other to forward packets to their destination. Routing protocols designed for the fixed network are not effective in mobile environments. Hence, a wide variety of routing protocols for mobile ad hoc networks (MANETs) have been designed and studied [4] [8]. The selection of a suitable ad hoc network routing protocol is crucial for a reliable file transfer.
- For the transfer, files should be split into several blocks. Therefore, receiver may temporally stops the transfer because of the likely disconnections with the sender. A mechanism for avoiding indefinited waiting for lost packets should be also provided.

Next sections aim at featuring the protocol behavior, showing performance results in real settings and concluding whether the protocol is suitable for CARLINK goals.

2 VDTP: Vehicular Data Transfer Protocol

VDTP provides a reliable file transfer service for VANET applications by using inter-vehicular communication (IVC). It is a connectionless protocol relying on LMR (*Lighweight Mobile Routing protocol* [3]), a reactive ad hoc routing protocol.

File transferring comprises two phases: information requesting phase and data requesting phase. When a file is being transferred, the node containing the file will be identified as **file owner** whereas the node requesting the file will be named **file petitioner**. On the one hand, file transferring is controlled by the file petitioner, according to the state diagram shown in Figure 1 (explained below). On the other hand, file owner is only in charge of replying petitioner requests. Moreover, file transferring is done by splitting the file into blocks with the same specified size.

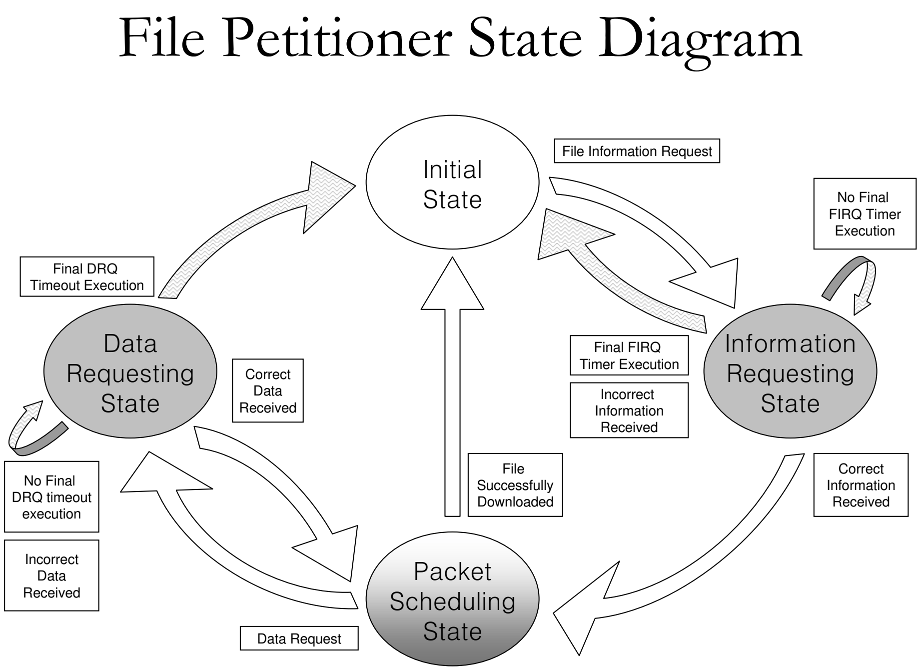

Figure 1: File Petitioner State Diagram

Communication between petitioner and owner is carried out by using the following packets: FIRQ (File Information Request), FIRP (File Information Reply), DRQ (Data Request) and DRP (Data Reply) (see Figure 2). When the file name and its location are passed to VDTP, a FIRQ packet is sent from the petitioner to the owner, in order to ask about the file size. The file petitioner then advances from **initial state** to **information requesting state** (Figure 1). The petitioner is now waiting for the file information coming from the owner. The file information will be hopefully received in a FIRP packet. Then, the file petitioner advances to **packet scheduling state**. It calculates the number of segments in which the file will be split, dividing the file size by the block size, and it also decides the next data segment to download. The petitioner requests this data segment by sending a DRQ packet to the owner and then advances to **data requesting state**. If possible, the data segment will be received in a DRP packet. Now, the file petitioner returns to **packet scheduling state** and repeats the same operation previously done. Note that each data segment is requested individually, and the next data segment is not requested until the current one is already received. When all data segments are downloaded, the file petitioner returns to **initial state**.

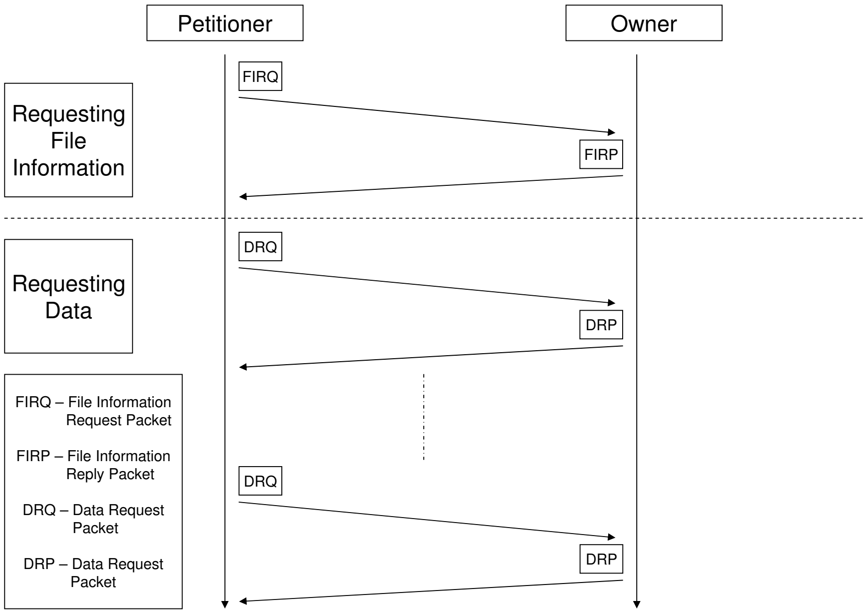

Figure 2: VDTP Time Diagram

Packet delivery is likely to fail due to the hostile communication medium. Therefore, VDTP provides a set of timers in charge of solving problems concerning packet lost or delay. The diagram is shown in Figure 3.

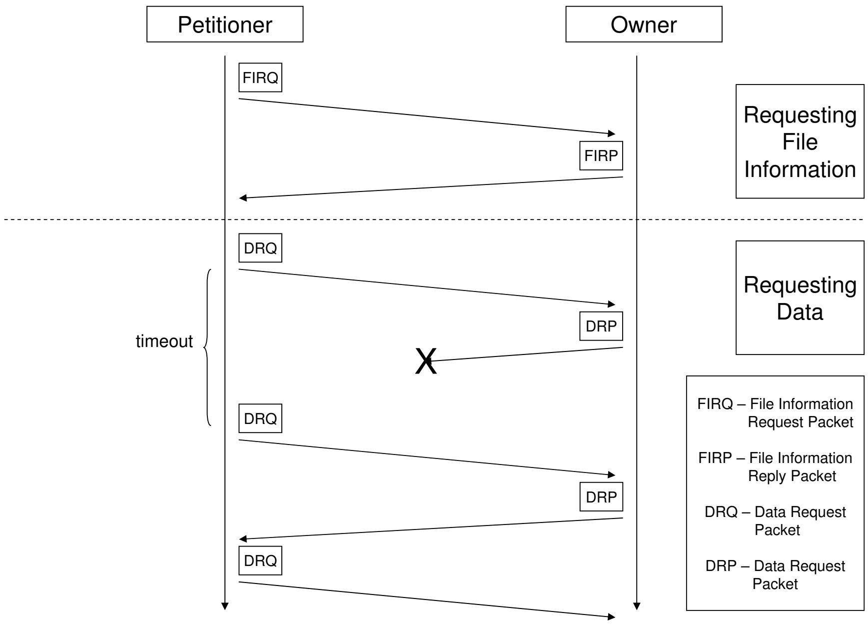

Figure 3: VDTP Time Diagram when DRP packet can not reach its destination

3 VDTP Performance evaluation

This section presents the performance evaluation of VDTP. Firstly, a detailed description of the real scenario is given as well as the hardware and software used. Afterwards, the experiment is described, the parameters to be measured are specified, and the obtained results are shown.

3.1 Real Scenario Description

We are currently interested in two simple scenarios. The first one, called **Scenario A** (see Figure 4), consist of two vehicles moving in a straight line with the same velocity (30 km/h). The total distance traversed by both vehicles is also the same (500 m). These vehicles are labelled as **file owner** and **file petitioner** denoting the role of each one in this scenario. The file petitioner is moving behind the file owner while the file transferring occurs.

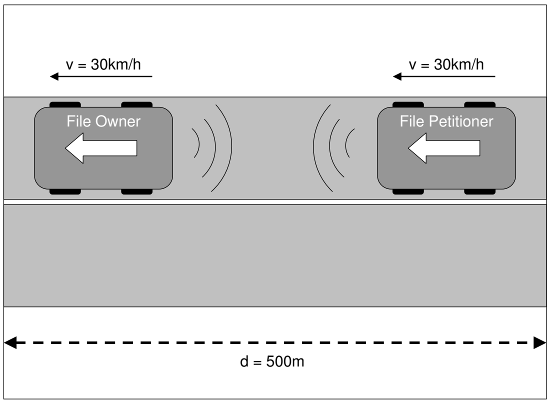

Figure 4: Scenario A

In the second scenario, called **Scenario B**, all parameters have the same value except the moving direction of the vehicles. In this case, they are also moving in a straight line but with opposite directions, as Figure 5 shows.

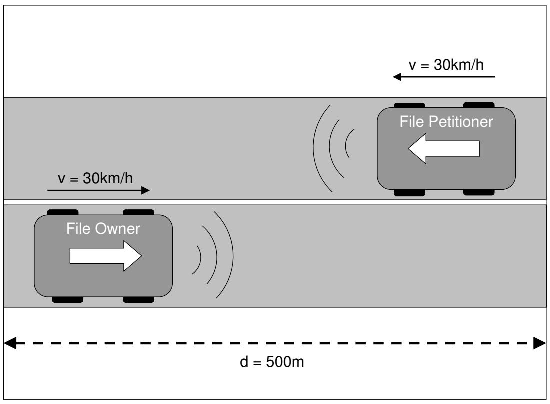

Figure 5: Scenario B

3.2 The Execution Platform

VDTP has been integrated into FSF (*Finding and Sharing Files* [7]), an application providing services for file searching and file sharing in MANETs. In this application, each user has a dedicated folder for placing files to be shared with all others. The users only need to enter a key-word for starting the file searching process. Then, FSF will broadcast search messages by using flooding techniques in order to reach the entire network. Once the users know the concrete file to be downloaded and the mobile devices where it is placed into, the file downloading process will be started by using VDTP.

FSF has been implemented with JANE [5], a Java-based middleware which is intended to assist ad hoc network researchers in application and protocol design. JANE aims at providing a uniform workbench supporting experiments ranging from pure simulation of mobile devices, over hybrid scenarios with interaction among simulated as well as real life devices, up to dedicated field trials as proof of concepts. Opposite to all studied VANET simulation tools [1], JANE allows to reuse the source code of simulated applications in real platforms without modifications. The advantages and disadvantages of using JANE are discused in [2].

The software mentioned above has been executed in laptops running the *Java Virtual Machine* of Sun Microsystems. These laptops were equipped with ORiNOCO[1] IEEE 802.11g compliant PC cards connected to a range extender antenna (8dB gain) in order to create an effective VANET.

3.3 The Experiment

The experiment is composed of several tests. Each test will consist of transferring a file in one of the previously specified scenarios (A or B). It is necessary to determine which type of files are representatives for CARLINK primary applications, i.e. what is the amount of data to transmit for each transfer. In this experiment we will use two different types: **file type 1** with 1 MB size (representing traffic information documents) and **file type 2** with 10 MB size (representing multimedia files).

The complete experiment will consist of realizing **ten repetitions** for each tests. The tests will be named as follows: **TestA1**, **TestA2**, **TestB1** and **TestB2**. The letter is describing the scenario and the number is denoting the file type.

3.4 Results

The results are presented in chronological order. It is important to remember that VDTP splits the file into chunks with the same size during the transfer. The chunk size can be configured manually in FSF. We selected the chunk size equal to 25 KB in all the tests.

Figure 6 shows the results of transferring ten times the file type 1 in scenario A. The average transmission time is 1.618 seconds, with an average transmission rate equal to 626.992 KB/s.

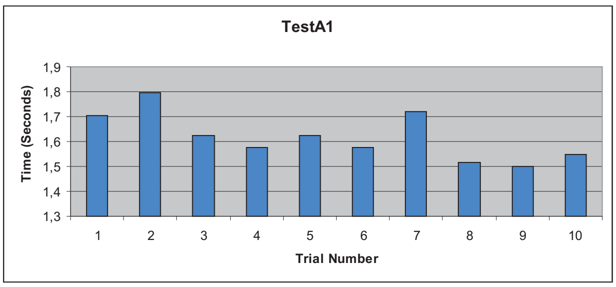

Figure 6: Transferring 1MB in Scenario A

[1] `http://www.proxim.com`

Figure 7 presents the results of transferring ten times the file type 2 in scenario A. The average transmission time is 17.328 seconds, with an average transmission rate equal to 585.176 KB/s.

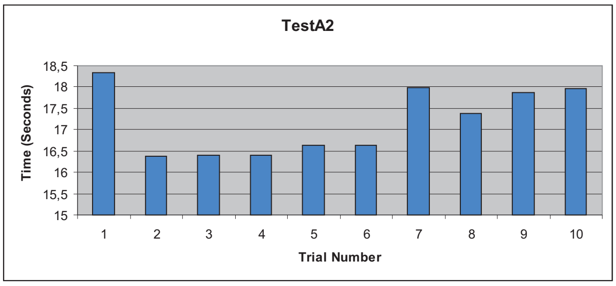

Figure 7: Transferring 10MB in Scenario A

Figure 8 shows the results of transferring ten times the file type 1 in scenario B. The average transmission time is 2.732 seconds, with an average transmission rate equal to 371.404 KB/s.

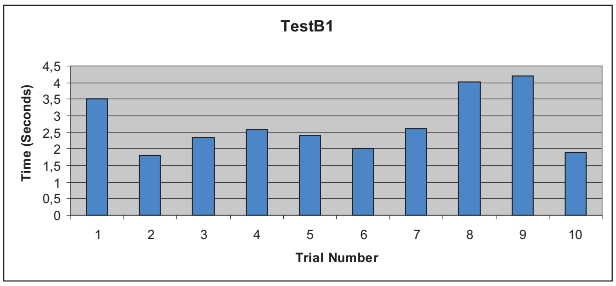

Figure 8: Transferring 1MB in Scenario B

Figure 9 presents the results of transferring ten times the file type 2 in scenario B. The average transmission time is 20.198 seconds, with an average transmission rate equal to 502.017 KB/s.

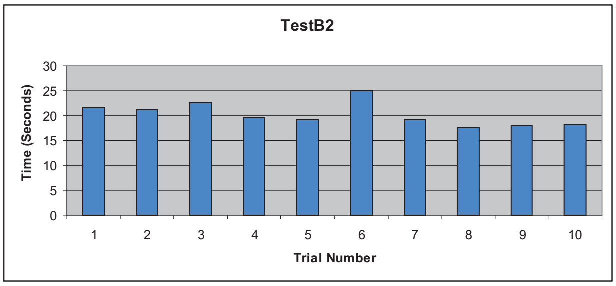

Figure 9: Transferring 10MB in Scenario B

Using 25 KB of chunk size implies to split the 1 MB file into 41 chunks and the 10 MB file into 406 ones. It is important to point out that the percentage of lost packets during all test was equal to 0%.

4 Conclusions

In this report, we have identified the main requirements for tranferring files in a reliable manner into vehicular ad hoc networks. We have implemented VDTP, a new protocol fulfilling those requirements, and it has been evaluated in real settings. The obtained results show that it is possible to perform file transferring by using inter-vehicular communications. In fact, it is possible to reach up to 625 KB/s of transmission rate, what provides a notion of the posibilities of the peer-to-peer communication paradigm inside VANETs.

The results allow us to conclude that VDTP is ready to be used in the CARLINK proyect. It is true that the real experiment has been carried out in simple scenarios. However, it is important to point out that most of the new communication protocols are presented nowadays by means of simulation solely. The use of JANE has been crucial in this aspect.

The next step is to perform advanced experiments. The first idea is to increase the number of involved vehicles in order to transfer files by using two-hops communication. We also would like to perform our test in more realistic scenarios such as highways. Once the reliability of VDTP is tested on advanced experiments, it could be also desirable to increase its performance by adding advanced techniques, such as sliding windows, instead of waiting for each packet's confirmation.

References

[1] CARLINK::UMA. Evaluating VANET simulators for CARLINK primary applications. Technical report, University of Malaga, Spain, 2007.

[2] CARLINK::UMA. JANE: A tool for implementing applications in real ad hoc networks. Technical report, University of Malaga, Spain, 2007.

[3] M. Scott Corson and Anthony Ephremides. A distributed routing algorithm for mobile wireless networks. *Wirel. Netw.*, 1(1):61–81, 1995.

[4] Laura Marie Feeney. A taxonomy for routing protocols in mobile ad hoc networks. Technical report, Swedish Institute of Computer Science, 1999.

[5] Daniel Gorgen, Hannes Frey, and Christian Hiedels. JANE - the Java ad hoc network development environment. In *40th ACM/IEEE Annual Simulation Symposium, ANSS 2007*, 2007.

[6] Alexander Klemm, Christoph Lindemann, and Oliver P. Waldhorst. A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. In *Proceedings of IEEE 58th Vehicular Technology Conference, 2003. VTC 2003-Fall*, 2003.

[7] Sebastián Luna. Designing and implementing ad hoc network applications by using JANE. Master's thesis, School of Computer Science Engineers. University of Malaga. Spain, November 2006.

[8] E.M. Royer and Chai-Keong Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 1999.