Master Thesis: Applying Delay Tolerant Protocols to VANETs

Oscar Trullols Cruces
trullols@ac.upc.edu

Advisor: Jose M. Barceló Ordinas
joseb@ac.upc.edu

Co-Advisor: Julián Morillo Pozo
jmorillo@ac.upc.edu

Master CANS
Computer Architecture Dept.
Tech. Univ. of Catalonia

30 June 2008
3.3.1 End-To-End ARQ ................................................... 28
3.3.2 Delayed Cooperative ARQ ........................................ 33
3.3.3 Experimental Prototype validation ................................. 35
3.3.4 End-to-End ARQ and DC-ARQ performance evaluation .... 39
3.3.5 Conclusions ....................................................... 47
3.4 Delay Tolerant Vehicular Protocol ................................. 48
  3.4.1 Carry-and-forward in opposite direction ....................... 49
  3.4.2 DTVP performance evaluation ................................ 51
  3.4.3 Conclusions ....................................................... 55

4 Summary, Conclusion and Future Work .......................... 56
Chapter 1

Introduction to Vehicular Ad hoc Networks

The cost reduction and fast evolution experienced by wireless communication technologies have made them suitable for a wide spectrum of applications. One of them is Vehicular Ad hoc Networks. We think that these wireless devices will have a high penetration rate among vehicles when being able to offer safety-related inter-vehicle communications and convenience and personalized applications.

Vehicular Ad hoc Networks are not just for fun, their aim is even to avoid accidents (e.g. using periodic broadcast of messages containing vehicles’ status information such as position and speed vector and a safety system aware of its surrounding to detect potential dangerous situations for the driver).

While safety applications could avoid injuries, convenience and leisure applications could increase the comfort of the driver and passengers: recommending routes depending on the traffic flow conditions, accessing Internet, gaming, sharing files or offering P2P services.

Vehicular Networks are characterized by

- high velocity of the vehicles
- environment factors: obstacles, tunnels, traffic jams, etc.
- determined mobility patterns that depend on source to destination path and on traffic conditions
- intermittent communications (isolated networks of cars due to the fragmentation of the network)
- high congestion channels (e.g. due to high density of nodes)
1.1 Identification of Challenges

The main challenges to achieve in this kind of networks are:

1. a scalable network that performs well with low and high node density scenarios,
2. copes with high speed nodes,
3. not suffers storm problems,
4. robust to hidden terminal problem,
5. not saturates the channel
6. and adaptive to the environment.

So, we may need to use a new set of protocols and architectures, based in the estimation of the trajectory of the vehicles, with the knowledge of velocity, position, direction and in an environment with harsh physical conditions (high losses).

Two set of performance parameters can be studied: "local" and "global" performance parameters. Local performance is related with each vehicle or transfer while Global performance is related with the whole network parameters. Average number of Access Points needed to download a file, average transfer delay and average throughput are local performance parameters. Number of vehicles able to download a file in an interval of time and fairness of the system are global performance parameters. The objective of this work is learning how Delay Tolerant Vehicular Protocols and mechanisms behave in terms of local and global parameters in a specific scenario.

1.2 Contribution

The beginning of this Master Thesis comes from two main ideas or objectives, (i) testing and evaluating the feasibility of a Delay Tolerant Network architecture in a Vehicular Ad Hoc Network in which vehicles access the backbone intermittently and (ii) testing how a Cooperative ARQ mechanism defined in the Thesis of Julian Morillo could be applied in this VANET scenario.

To evaluate these objectives, we first tested experimentally a prototype of DC-ARQ to observe that this mechanism behaves well and improves loss packets ratio. This work was published in [1].

After the experimental prototype, we evaluated DC-ARQ using simulation to allow a bigger scene with more nodes and different car densities. This work was published in [2].

Following we simulated this protocol in a whole VANET architecture, where we had to define an opportunistic carry and forward mechanism.
As a summary, the main contribution of this Master Thesis is:

1. Definition and evaluation of an end-to-end ARQ mechanism to offer a coordination and retransmission service that fits this Delay Tolerant scenario.

2. Evaluation of the Delayed Cooperative ARQ (DC-ARQ) scheme to be used in vehicular networks where cars download delay-tolerant information from AP on the road and can use the areas without AP connectivity to cooperate and request the retransmission of lost frames to other cars in the platoon. The definition of DC-ARQ is not product of this Master Thesis, only the adaptation to a VANET scenario.

3. Definition and evaluation of a Vehicular framework (DTVP) that in addition to the Delayed Cooperative ARQ, uses packet transfers scheduled from vehicles coming in opposite direction.

To evaluate the scheme we have simulated using ns-2 [3] the case of a straight road with sparse AP giving intermittent connectivity to vehicles, like Figure 1.1.

Figure 1.1: The intended scenario consists on a road with sparse APs
Chapter 2

Background and Related Work

2.1 Vehicular Ad hoc Networks

2.1.1 Introduction to VANETs

Mobile Ad Hoc Networks (MANETs) are characterized as networks in which nodes, static or mobile, act as a host and as a router extending the one-hop coverage area of a single wireless network. These networks are self-organized, and typically nodes follow random mobility patterns. It seemed that MANETs market penetration would grow with the fast introduction of IEEE 802.11 standards. However, applications using MANET technologies have not massively impacted the market. Pure general-purpose MANET lacks realism from a technical standpoint. Channel impairments highly impact protocol performance parameters and mobility models used in MANET research do not reflect real-world situations. There is a general belief that researchers did fail in providing implementation and integration of solutions (e.g. system prototypes), M. Conti and S. Giordano states possibles causes in [4] and [5]. However, there are some examples of Ad Hoc networks that have a market when approached in a pragmatic way and in which industry and operators have an interest. Not pure general-purpose MANETs examples are: Mesh Networks, Opportunistic Networks, Sensor Networks and Vehicular Networks.

Vehicular Ad Hoc Networks (VANETs) are a kind of MANETs in which nodes are vehicles that follow particular mobility patterns regulated by vial normative. The concept of Hybrid MANETs appears when relaxing the constrain of not having infrastructure (e.g. VANETs with Vehicular-to-Road communications).

Communications in VANETs may be of two types: single-hop to neighbor cars to advise of an event (e.g. braking) or multi-hop to either disseminate information or to query for a service. Applications can be categorized as Transportation-related applications, those applications that increase the safety of the driver and passengers, and Convenience and personalized applications, those applications that increase the comfort of the driver and passengers.

Several organizations are supporting standardization activities applied to ve-
hicular communications. Examples of these activities are IEEE 802.11p and WAVE (Wireless Access Vehicular Environment), ISO TC204 WG-16 CALM architecture, ETSI-ERM TG37. Other organizations such as C2CCC (Car to Car Communications Consortium) have brought the European vehicle manufacturers to join in an initiative with the objective of further increasing road traffic safety and efficiency by means of inter-vehicle communications.

As in MANETs, communications in VANETs may be impacted by several factors:

- high velocity of the vehicles
- environment factors: obstacles, tunnels, traffic jams, etc.
- determined mobility patterns that depend on source to destination path and on traffic conditions
- intermittent communications (isolated networks of cars due to the fragmentation of the network) and possible need of efficient geo-casting and flooding mechanisms
- high congestion channels (e.g. due to high density of nodes)

### 2.1.2 Types of Applications in VANETs

Transportation-related applications are those applications that increase the safety of the driver and passengers. Transportation-related applications range from safety applications such as cooperative forward collision warning or extended electronic brake lights to traffic management applications such as road-condition warnings or alternative route warnings. Convenience and personalized applications increase the comfort of the driver and passengers. These applications range from Internet access, gaming, sharing files or P2P services.

Typical safety applications are characterized as being applications in which the main objective is to disseminate certain event that have occurred in the vicinity of the sender. Some examples described in [6]) are:

- *Cooperative awareness*: to extend non-cooperative driver assistance systems which perception is limited to the operative range of on-board sensors (adverse weather, obstacles or dangerous road conditions).
- *Cooperative assistance*: distribution of data (e.g. warning of accidents).
- *Cooperative manoeuvring*: exchange of relative position and dynamics between vehicles (e.g. Lane Merge/Lane Change Assistance, Adaptive Cruise/Cooperative Driving)

Typical Convenience and personalized applications are:
• **Car to Mobile devices:** those applications between the car and mobile devices (e.g. mobile phone, MP3, laptop, etc).

• **Car to Home and Car to Office:** communications between the car and private networks either at home or at office.

• **Car to Enterprise:** communications between the car and companies (e.g. restaurants, gas stations, parking areas, etc) that give road services.

• **Car to infrastructure:** information received by a car from hot spots giving road and traffic information and car access to Internet.

• **Car to Car:** exchange of information between car users (e.g. files, platoon traveling, etc).

Transportation-related applications and Convenience and personalized applications may establish communications between vehicles, named V2V (Vehicular to Vehicular) communications or between a vehicle and road infrastructure, named V2R (Vehicular to Road) communications. V2V communications are also called C2C (Car to Car) and V2R communications are also called C2R (Car to Road).

The type of requirements that demand each communication depends on each type of application. However, there is a set of requirements that is common to almost all applications, see [7], where is described a range of VANET application requirements. The most interesting requirements are: coverage should be in the range of 10 to 1000 meters with a car maximum relative speed of 500 Km/h. Latency ranges between 50 ms to 500 ms. In general, safety applications should not wait more than 200 ms. Network density may range from small platoons of 2 to 20 cars, to traffic jams with around a thousand of cars per radio cell.

In general, safety applications are characterized by being transactions that range data packet sizes from 100 Bytes to 2 KBytes and that have to reach a few number of hops, most applications being for a single hop. A likely situation may arise in case there are traffic jams and redundant packets of multiple nodes consume the bandwidth. Therefore, beacon-based protocols should be designed to control the beacon frequency and should find a trade-off between beacon frequency, latencies and bandwidth usage. Safety applications should therefore use different channels than comfort applications and MACs should take into account the problem of the hidden terminal and the exposed terminal since they may be disastrous for safety applications.

Two kind of safety messages may appear in safety applications: periodic messages and event-driven messages. Periodic messages are sent with the intention of detecting non-safe situations such as providing information (e.g. position, speed, direction, etc) about surrounding vehicles. An interesting point discussed in [8] is the transmission power control. Increasing the transmission power improves radio channel communication and higher area of reachability. However, high transmission power also means higher probabilities of collision (e.g. in high node density.
areas). In safety applications, there should be fairness between different applications and nodes. Event-driven messages, see [8], are those messages generated on demand. Examples are an alert in a dangerous situation (e.g. front car braking) or a request in a routing protocol. In these type of messages, rapid diffusion of the information is vital (e.g. to avoid an accident).

### 2.1.3 Radio Spectrum and Standards of VANETs

Intelligent Transport Systems (ITS) will use the 5 GHz band spectrum although there still are some discrepancies in the channel allocation between several countries and organizations. Architectures using DSRC (Dedicated Short Range Communication) have a dedicated band for ITS services ranged from 5.885 to 5.905 GHz in USA and from 5.795 to 5.815 GHz in Europe. Furthermore, the ITU has requested for ITS Safety applications the allocation of 75 MHz from 5.850 to 5.925 GHz with the idea of supporting both 10 and 20 MHz channels. IEEE 802.11a works on the unlicensed bands 5.15-5.25 (USA UNII lower band), 5.25-5.35 (USA UNII middle band), 5.470-5.825 (USA/Europe) and 5.725-5.825 (USA UNII upper band). IEEE 802.11p WAVE, being part of the DSRC system, operates in the licensed 5.9GHz band. Figure 2.1 presents the radio spectrum allocation in the 5 GHz band. Robust safety vehicular communications need a protected frequency band outside the unlicensed ISM band.

![Figure 2.1: 5 GHz Band Spectrum](image)

There is an effort pursued by several organizations to find a common architecture for VANETS. As a technology, DSRC will use IEEE802.11p as background technology, although C2CCC stands for a modified European IEEE802.11p version based in the ETSI channel allocation. ISO/CALM pursues a VANET architecture able to communicate continuously in any technology and based on IETF mobile protocols (e.g. Mobile IPv6) with IPv6 as network protocol. C2CCC pursues an architecture based on IEEE802.11p technology and without any requirement at
upper layers (e.g. they consider network technologies such as TCP/IP).

IEEE802.11p, also called WAVE (Wireless Access in Vehicular Environments), is a multichannel wireless standard based on IEEE802.11a PHY standard and IEEE802.11 MAC standard (i.e. WAVE uses CSMA/CA as basic medium access mechanism). WAVE would use one control channel to set up transmissions and data channels to send the data. WAVE allows high data rate (≤ 27 Mbps) in short distances (≤ 1 Km). There are some differences with IEEE802.11a. IEEE802.11p, (i) operates in licensed bands at 5.9 GHz, (ii) with 7 channels supporting safety and non-safety applications, (iii) in a 10 MHz channel bandwidth and (iv) is thought for outdoor high speed vehicles.

### 2.1.4 PHY layer in VANETs

The wireless radio channel causes a great impact in the reception of packets. Path loss and shadowing cause the variation in received signal power over distance. Path loss, [9], is caused by dissipation of the power radiated by the transmitter as well the effects of the propagation channel. Shadowing is caused by obstacles between transmitter and receiver that attenuate signal power through absorption, reflection, scattering and refraction. Variations due to path loss occurs over long distances while shadowing occurs over distances proportional to the obstructing length. Since both are relatively long distances they are considered as large-scale propagation effects. Multipath is due to the receiving of multiple components of the signal. These components may be delayed, attenuated, shifted in phase and/or frequency from the LOS (Line of Sight) signal path at the receiver. Variations due to multipath are on the order of the wavelength and are considered as small-scale propagation effects.

There are different models for signal propagation between the transmitter and the receiver. Some models considered in MANET studies are:

- The **Free Space Model** that considers a perfectly reception of the signal over one path at distance $d$. The receptor is on Line of Sight (LOS) and free of obstacles.

  \[
  P_r = \frac{P_t G_t G_r \lambda^2}{(4 \pi)^2 d^2 L} \tag{2.1}
  \]

  where $P_r$ and $P_t$ are the receiving and transmitting power, $G_r$ and $G_t$ are the receiving and transmitting antennas gains, $\lambda$ is the wave length, $L$ is the system loss and $d$ is the distance between receiver and transmitter.

- The **Ground Reflection Model** considers that ground reflection dominates the multipath effect. An examples of this model is the 2-Ray model:

  \[
  P_r = P_t G_t G_r \frac{h_r^2 h_t^2}{d^4 L} \tag{2.2}
  \]

  where now the new parameters $h_r$ and $h_t$ are the heights of the receiver and transmitter antennas. The model neglects obstacles.
2.2 Delay Tolerant Networks

- Shadowing models handle variations due to shadowing. In the probabilistic shadowing model a Gaussian random variable \( \chi \) is added to the Long Distance Path Loss model to reflect the variations of the received power at a given distance:

\[
P_r|_{dB} = P_{r0}|_{dB} - 10\beta \log\left(\frac{d}{d_0}\right) + \chi|_{dB}
\]  

(2.3)

For outdoor free space urban environments the path loss exponent \( \beta = 2 \), and for outdoor shadowed urban areas \( 2.7 \leq \beta \leq 5 \). \( \chi \) is a Gaussian variable with zero mean and \( \sigma^2 \) variance. Typically values of shadowing \( \sigma_{dB} \) are between 4 to 12 for outdoor.

Another probabilistic model used for modeling radio channels is the Nakagami model. In this model the received power is calculated using the following distribution:

\[
P_r = \frac{2}{\Gamma(m)} \left(\frac{m}{\Omega}\right)^m r^{2m-1} e^{-\frac{m r^2}{\Omega}}
\]  

(2.4)

where \( \Omega = E(R^2) \) denotes the second moment of random variable \( R \) (the amplitude of the received signal), \( \Gamma(m) \) is the Gamma function, and the fading parameter \( m \) is defined as \( m = \frac{\Omega}{E[(R^2 - \Omega)^2]} \geq 1/2 \).

- Empirical Path Models are based on measurements over a given distance in a given frequency and a particular area. (e.g. The Okamura-Hata Model considers the effects of diffraction, reflection and scattering caused by city structures, covers frequencies from 150 MHz to 1500 MHz and transmitter heights up to 200m)

M. Torrent et al, in [10], uses the Nakagami distribution as radio channel model to study channel access times and reception probabilities in a highway scenario. For that the authors collect empirical data from radios mounted on vehicles moving on highways. Figure 2.2 shows the shape of a 2-Ray-ground model and the Nakagami model (shape taken from figure 4 of paper [10]) in absence of interference.

The Physical layer chosen have a severe impact on the simulations. Takai et al [11] compare the effect of different PHY parameters and models (SNR calculation, Signal reception, fading and path loss) on routing protocols AODV and DSR, using several network simulators (ns-2, glomosim and OPNET). The paper shows the importance of the PHY layer on performance parameters at higher layers.

2.2 Delay Tolerant Networks

2.2.1 Introduction to DTNs

As stated before, Mobile Ad Hoc Networks (MANETs) are characterized as networks in which nodes, static or mobile, act as a host and as a router extending
2.2. Delay Tolerant Networks

the one-hop coverage area of a single wireless network. Packets traverse an ad hoc network by being relayed from one node to another until they reach their destination. Because nodes are moving, the topology of the network is in constant change, and to find the destination and the route could be hard challenges. Routing in mobile ad hoc networks is a well-studied topic, several routing protocols have been proposed, such as OLSR [12], AODV [13], DSR [14]. But they would not work in delay tolerant networks because they assume:

- An end-to-end path between any pair of nodes.
- Small maximum round-trip time between them.
- Small hop-by-hop and end-to-end packet drop probability.

Delay tolerant Networks must be built having all of these characteristics in mind.

2.2.2 Types of Applications of DTNs

Examples of Delay Tolerant Networks are:

- Mobile Networks: due to mobility, networks unexpectedly, periodically or predictably may become partitioned. In these networks is very useful to know the behavior of the movements, to propose Model-Based approaches that will fit better.
- Sensor Networks: characterized by lots of nodes with high limitations to power, memory, and/or CPU capability. These networks use Scheduled-Based approaches to allow the nodes to conserve power using periodic sleep times.
- Military Ad-Hoc Networks: high challenged networks due to high velocities, radio jamming, higher requirements of QoS or prioritized services.
• Remote rural communities: connecting end-users at remote-sites could be not economically feasible. Distribute content to these places could be possible installing wireless information "kiosks" in these villages and leveraging existing communications and transportation infrastructure.

• Pocket switched networks: make use of human mobility in order to transfer data between mobile users’ devices.

• Exotic Media Networks: include free-space and communication in water. Are subject to high propagation delays, predictable interruption (e.g. a passing satellite), or may suffer outage due to the weather.

2.2.3 Characteristics of DTNs

Delay tolerant networks are characterized by latency, bandwidth, error probability or path stability limitations that are worse than the characteristics of the common today’s Internet. As a baseline, we will be comparing with common Internet performance

Network partitions: End-to-end disconnection may be more common than connection. Disconnection could be consequence of a fault or not. Non-faulty disconnections could arise from predictable(satellite passes) or opportunistic motion(random walk nodes) and probably predictable sleep times (sensor networks). DTN require the routing protocols to understand that the lack of reachability may be the result of a normal situation and should not be considered a network operation fault.

High latency, long queuing times: The challenging medium affects the transmission and propagation delays. Data rates may be asymmetric or even unidirectional. On the common networks, queuing time rarely exceeds a second and if next-hop neighbors are not instantaneously reachable packets are discarded. On DTN, queuing time could be extremely large (minutes, hours, or even days), messages may be stored for long periods of time at routing nodes.
2.2. Delay Tolerant Networks

**Uncertain Error probability:** On common networks, links could offer a delimited error probability, while at DTN the error probability of a link may change unexpectedly.

**Limited Devices:** In some kinds of delay tolerant networks nodes with highly restricted power, memory or processing capability are used, due to price, technical or size limitations.

2.2.4 Weaknesses of common Internet protocols at DTN

To face the DTN challenges we could study which features of the common and experienced Internet protocols will fit in the delay tolerant scenario and which features should be avoided. This is based on a previous work of Kevin Fall [16]. The mentioned characteristics of these networks contribute to confound the efficient operation of the common Internet protocols.

At the **Transport Layer**, in the case of using TCP, high latency and error probability interfere with connection establishment, it is a “chatty” protocol and if one step’s packet is lost, the connection establishment fails. Even if this issue is solved, when multiple loss events occur it continues retrying with a backed-off retransmission timer until giving up and terminating a connection, that will require to restart the establishment again. Even with low path loss, it highly affects the performance invoking the fast retransmit and recovery algorithms, because assumes that unacknowledged segments are due to network congestion. While this is an acceptable assumption for many networks, in delay tolerant networks segments may be lost for other reasons, such as poor data link layer transmission quality.

At the **Network Layer**, in the case of using an IP based protocol, performance could also be affected by path loss. Datagrams may be fragmented in several pieces. But IP does not have any mechanism for fragment retransmission, then the probability of successful datagram delivery would be further reduced.

Related to **Routing protocols**, the DTN’s characteristics would affect the correct operation of routing protocols, because they continuously try to compute paths of the whole graph from the network information received, and if path loss is high, the interpretation of this information could change suddenly even if the topology have not changed.

At the **Application Layer**, electronic mail has interesting features that are expected for a DTN, like the asynchronous type of message delivery system or the flexible name/addressing semantics. But it lacks of a dynamic routing mechanism and it has a weakly-defined but possibly useful error reporting features.

Most of the problems come from the Internet’s idea of fate sharing. One of the Internet basis’ idea was that the per-connection state should remain only in the ends, to avoid that a failure in any of the forwarding nodes of the path could heavily impact the connection. In some scenarios of DTNs, it would be very useful to allow the end nodes to delegate their end-node connection state to other nodes.
2.2. Delay Tolerant Networks

Based on Internet experience, the DTN system should have an overlay routing capability, similar of that used in peer-to-peer systems, combined with a delay an disconnection tolerant properties similar of that used by electronic mail.

2.2.5 DTN architecture

The RFC4838 (Delay-Tolerant Networking Architecture [17]) proposes a general architecture to overcome all the previous challenges based on store-and-forward message switching. It is based on asynchronous messaging and uses postal mail as a model of service classes and delivery semantics. It introduces a novel approach to end-to-end reliability across frequently partitioned and unreliable networks and proposes a model for securing the network infrastructure against unauthorized access.

Blocks of user data, called bundles, are forwarded from the source to a storage on another node, that assumes the responsibility for reliable delivery of the bundle to its destination, these nodes are called custodians. The bundle would travel along a path of custodians that eventually reaches the destination. Figure 2.4

![Custodians may store bundles for long times in persistent storage](image1)

**Figure 2.4:** Custodians may store bundles for long times in persistent storage

**Bundles:**
The bundle layer, see Figure 2.5, over the transport layer provides a store-and-forward forwarding service using persistent storage, and provides functionality similar to the Internet layer of gateways. It differs from them because it is layer-agnostic and is focused on bundle forwarding rather than packet switching. Providing interoperability between underlying protocols specific to each environment.

![DTN Layers](image2)

*Figure 2.5:* DTN Layers

**Custodians:**
Custodians allow the source, or the previous custodian, to free retransmission-related resources relatively soon because the next custodian gets the retransmission
responsibility. Not all the nodes in a DTN are required to accept a custody. Having a custody means you need to share some retransmission-related resources maybe for a long time, and some nodes may have not sufficient resources to do this and would only act as forwarders.

![Figure 2.6: Custody may not be hop-by-hop](image)

2.3 Delay Tolerant VANETS

2.3.1 Routing protocols in Delay Tolerant VANETs

One of the main researched topics on MANETs have been Routing protocols. These protocols may be classified according to the following categories:

- Topological routing protocols
- Position-based routing protocols
- Energy-aware routing protocols
- Hierarchical routing protocols
- Opportunistic routing protocols

Not all of these protocols are appropriate to Delay Tolerant VANETs. For example Topological routing protocols are not adequate due to the fast mobility of nodes that produces fast varying network topologies. Energy-aware routing protocols are not related to vehicular networks because energy consumption is not a problem in a vehicle. Hierarchical routing protocols have been studied in the context of clustering nodes that implement a topological routing protocol for intra-cluster routing and another topological routing protocol for inter-cluster routing. The same ideas may be applied to VANETs using other classes of routing protocols that are not topological. Position-based and opportunistic routing protocols seem to be good choices in VANETs, due to the fact that vehicles may incorporate GPS devices able to provide position parameters and the intermittent nature of vehicular networks. There is another category of protocols that seem interesting to VANETS: routing based on trajectories. This kind of protocols are based in the estimation of the trajectory of the vehicles with the knowledge of velocity, position and direction.
2.3. Delay Tolerant VANETS

2.3.2 Broadcast and dissemination techniques

In a push communication model, information is exchanged through two different mechanisms: (i) flooding and (ii) dissemination. In **flooding mechanisms**, a node propagates information to neighbors. These neighbors retransmit each broadcast packet received. The objective of the broadcast is that each packet is received by each node of the network. The differences between broadcast mechanisms is how to reach all the nodes with the minimum number of retransmissions. This is equivalent to choose the minimum number of forwarders to reach the whole network. In **dissemination mechanisms**, a node broadcast its own information and the information of other nodes. A node receiving a dissemination packet, *merges* the received information with its own information and retransmits the updated information in the next broadcast period.

Broadcast packets in a network may result in high redundancy, contention and collisions. In general, eliminating redundancy is addressed through the use of topology information. Contention is due to the fact that several nodes decide to re-broadcast the packet. The solution is a forwarding mechanism that uses the channel more efficiently minimizing the number of nodes that contend for the media and the number of re-broadcasted packets reaching the whole network. It must be noticed that RTS/CTS mechanism decreases the effect of the hidden terminal and therefore collisions while the ACK makes the channel more reliable. However, when broadcasting a packet the RTS/CTS is deactivated and there are no ACKs.

Paper [18] surveys and compares different broadcasting techniques to forward packets. The authors classify broadcasting mechanisms in four classes:

- **Simple Flooding**: each node retransmits the packet so packets are flooded to the whole network. Simple mechanism at the cost of maximum number of retransmissions per packet.

- **Probability-Based Methods**: in the Probabilistic scheme packets are retransmitted using certain probability, so, no all nodes retransmit each packet. Probabilistic schemes behave well in dense networks. However in sparse networks there will be packets not received unless the probability tends to 1.

- **Counter-based Methods**: in the counter based schemes the nodes initiate a counter to one and increase the counter upon reception of each copy of the packet. After a RAD (Random Assessment Delay) time, the packet is retransmitted if the counter is less than a threshold. Under this scheme there will be nodes not retransmitting in dense networks while all nodes will tend to retransmit in sparse networks.

- **Area-Based methods**: try to maximize the additional coverage area that will cover the node that re-broadcasts the packet. However, nodes have not knowledge of other nodes existing in the covered area.
2.3. Delay Tolerant VANETS

- **Neighbor-Knowledge Methods**: are based on the knowledge of 1-hop and 2-hop neighbors. In general, these mechanisms are hard to apply in a VANET environment with high mobility in which it is difficult to keep updated the knowledge of the neighborhood.

It is difficult to find broadcast mechanisms easy to implement, that copes with high speed nodes, that not suffers storm problems, robust to hidden terminal problem, that not saturates the channel and maybe adaptive to the environment (e.g. node speed, node density). There also is a need to prioritize among relevant safety messages and the mechanisms have to be efficient in high node densities (e.g. traffic jam).

### 2.3.3 Topological routing protocols

Set of protocols that use information about the links (metrics such as next-hop, bandwidth, etc) to perform packet forwarding. **Proactive Algorithms** (such as DSDV or OLSR) build the routing table obtaining information about the links, even when these links are not used. DSDV is a distance-vector protocol, while OLSR is a link-state protocol. **Reactive Algorithms** (such as DSR or AODV) build the routing table dynamically on demand. In general, Proactive protocols have small delays since they have the entries built at the cost of higher bandwidth consumption due to the frequent control packets needed to maintain the routing tables. In general, reactive Protocols have longer delays due to the fact of having to look for a route at the moment of the connection. However, depending on specific parameters (e.g. mobility patterns or traffic load) any of these approaches may behave better or worse. There are many surveys on Topological routing for MANETS (e.g. see [19]).

There have been works in VANETs in which some of these protocols, mainly AODV, have been studied. Paper [20] compares two position-based algorithms with AODV, see next section 2.3.4. The packet delivery ratio using AODV as a function of the distance (hops) decreases almost to zero after 7-8 hops (4000m in a communications range of 500m) and packet delivery ratio of 0.2 after 4 hops (2000m in a communications range of 500m).

### 2.3.4 Position-based routing protocols

Set of protocols that need the position (e.g. via GPS) of the participating nodes. In general, these algorithms look the position of the destination nodes using a Location Server and add this position in the packet header. Nodes that receive the packet apply a forwarding strategy to retransmit the packet. Each node stores a node ID, the direction and distance to the node, as well as an age time.

Forwarding Strategies decide towards which node or area the packet is forwarded. **Greedy Packet Forwarding** forwards the packet to a neighbor lying in the direction of the destination. Some examples (see [21]) are: (i) **Most forward**
within $r$ (MFR) that forwards packets towards the node that makes more progress towards destination. (ii) *Nearest with forward progress (NFP)* that forwards packets towards the node that is nearest the source and closer to destination. (iii) *Compass routing* selects the neighbor closest to the straight line between sender and destination. (iv) *Random forwarding* chooses randomly one the neighbors closer to the destination than the sender.

Paper [20] compares two Greedy Packet Forwarding strategies: Position-Based Forwarding (PBF) and Contention-Based Forwarding (CBF) in a vehicular scenario. *Position-Based Forwarding* uses a location service to learn about the current location of the destination and then applies MFR as a forward strategy. *Contention-Based Forwarding*, does not make use of beacons, but the forwarder uses a contention period in which each node selects a waiting time depending on the distance to the destination. It shows how CBF benefits from not pre-selecting a forwarder node with the Nakagami channel model. This is due to the fact that using this stochastic channel model, nodes outside the communication range may receive packets and are able to decode them.

### 2.3.5 Opportunistic Routing

New routing techniques have to be devised to take into account sparse and intermittent networks in which nodes communicate either scheduled over time or randomly. Pelusi et al in paper [22] give a review of opportunistic techniques in ad hoc networks. In an opportunistic network, each node decides locally to which next hop the packet will be forwarded. This next-hop may decide to store the packet until a new opportunity to forward the packet appears.

**Forwarding and Replication Strategies**

Wired routing protocols and wireless topological routing protocols choose next hop neighbors among those that have a path towards destination with a lower cost with respect to a metric or a set of metrics. In opportunistic networks in which there may not be a path to the destination, choosing a next hop based on global topological cost metrics is not possible. The forwarding of packets has to be based on *carry and forward* techniques that impact the performance of the routing protocol and the reliability in delivering a packet. Two main philosophies define the way of designing *carry and forward* algorithms: *replication strategies* replicate copies of each packet until one of them gets destination. *Forwarding strategies* maintain only one copy of the packet in the network. We begin with replication strategies and with the epidemic algorithm as its main representative.

Epidemic algorithms, see [23], state that given random exchange of data among replicas, all updates will be seen by all replicas in a bounded amount of time. In Ad hoc networks, this idea translates to deliver a packet with high probability to a particular host. Any host will buffer its own generated messages as well as messages belonging to other hosts that previously met. Each host maintains
a cache of hosts that has met recently. When a host met another host the two hosts exchange a summary vector to determine which messages stored remotely have not been seen by the local host. Then, hosts request packets not in their cache. A key feature in epidemic routing is that hosts may decide which packets to accept (e.g. the host may establish limits in the number of packets, size of packets, etc). Paper [23] models a maximum queue size as a limiting policy in the number of packets a node is willing to carry on behalf of other nodes and a hop count per packet as a policy in the maximum number of epidemic exchanges that a particular packet is subject to (e.g. hop count equal to one only will be delivered to the destination). The main disadvantage of epidemic routing is the flooding of packets that consumes network resources.

T. Spyropoulos et al, in [24], propose Spray and Wait, that “sprays” a number of $L$ copies into the network and then “waits” that one of these $L$ nodes carrying the message performs direct transmission with the destination. The idea behind delivering multiple-copies against single-copies is to provide high probabilities of delivery and low delays at the cost of increasing memory resources. Different ”spraying” heuristics are defined: Binary Spray and Wait assumes that instead of a source giving a copy to $L$ different nodes (called Source Spray and Wait), the source, each time that it encounters a node, hands over ⌊$n/2$⌋ copies and keeps ⌊$n/2$⌋ copies for itself, when it is left with only one copy, it switches to direct transmission. Spray and Wait attempts to reduce average message delivery delay. Although does not take into account memory or bandwidth constraints. T. Spyropoulos et al conclude that replicating may offer better performance than just forwarding, see [24].

S. Jain, in [25], propose a framework for evaluating routing algorithms in delay tolerant networks. Their proposal belongs to the Forwarding Strategies set. Jain et al gives a definition of contact as an opportunity to send data over an edge and a corresponding interval during which the edge capacity is strictly positive (when no communication is possible, the edge is assigned a zero capacity). The routing evaluation framework is developed around the concept of knowledge oracles. These oracles are notational elements used to encapsulate particular knowledge about the network required by the different algorithms. Examples of oracles are (i) Contacts summary oracle that provides the average waiting time until the next contact for an edge, (ii) contacts oracle that answer any question about contacts between two nodes at any time, (iii) queueing oracle that gives information about instantaneous buffer occupancies and (iv) traffic demand oracle that answer questions regarding the present and future traffic demand.

Paper [26] uses the concept of passive cure in which the destination sends a cure-ack to the last forwarder to indicate that the packet has reached destination. The cure-ack ”heals” the forwarding of relaying the packet more times and diminishes the total number of packets in the network. The healing propagates through the network preventing future retransmissions of the same packet.

A similar work presented by T. Small, in [27], and called SWIM (Shared Wireless Infostation Model), controls the probability of packet transmission be-
2.3. Delay Tolerant VANETS

tween two adjacent nodes (probability of infection in an epidemic model), the transmission range of each node (infection distance in an epidemic model) and number and distribution of Infostation (number and location of the hospitals in an epidemic model). SWIM assumes a network of whales moving in oceans in which there are placed SWIM station (e.g. buoys). Each whale transports its own data and the data of the whales encounter in its way. The propagation of each packet of information generated by a whale is modeled as spread of one infectious disease. When the whale meets a infostation delivers its data and ”heals” from the disease.

A. Lindgren, in [28], propose PROPHET, a Probabilistic Routing Protocol using History of Encounters and Transitivity and compare it with Epidemic Routing. The main idea behind probabilistic routing is that in some situations locations are frequently visited in the past, so, with certain probability these locations will be again visited. PROPHET works very similar to Epidemic Routing. Summary vectors contain the delivery predictability information stored at the nodes. This delivery predictability states the probability to meet each node in the network based on previous encounters with that node. In this way, a node \( B \) that meets a node \( A \), which has a packet to a destination, after receiving \( A \)'s summary vector may decide whether to copy the packet given that has more probability to encounter the destination node than the node \( A \).

In general, flooding as Epidemic Routing has a huge performance impact (e.g. overhead) in the network. Probabilistic Routing reduces the amount of traffic since nodes chose those relays that have higher probability to meet destinations. Another approach to reduce overhead is to use a single copy mechanism. In this way the copy goes through different nodes that may be better positioned to meet the destination at the cost of increasing delay and delivery success rate. Multi-copy strategies plays with the trade off between delay and overhead. The idea behind multiple-copies strategies is to define a Replication Factor (RF) or number of copies to be relayed per packet. The way in which the copies are delivered determine performance parameters such as delay and delivery ratio. Finding a suitable RF is difficult and in general depends of mobility patterns, number of nodes and network topology.

**Opportunistic Routing in VANETs:**

J. LeBrun et al, in [29], study five opportunistic forwarding strategies, included MoVe, an algorithm which uses velocities to make intelligent forwarding. All of them use a Hello-Response technique to detect mobile nodes. The approaches are: (i) NoTalk (similar to Data Mule project) in which nodes accept data from a source and carry the data until meet the destination, (ii) Broadcast in which nodes exchange data with every other node it meets, (iii) Location-based in which nodes forward data to neighbors only if that neighbor is closer to the destination than its own current position, (iv) MoVe in which nodes leverage relative velocities and predict who is the closest to destination, and (v) MoVe-Lookahead in which
2.3. Delay Tolerant VANETS

nodes follow the MoVe rules but taking into account node trajectory changes. The authors use simulation in a 4Km x 4Km terrain but they also experiment with a network of buses in the San Francisco MUNI System.

J. Zhao et al, in [30], presents VADD (Vehicle-Assisted Data Delivery) protocols for forward packets to the best road, in urban scenarios, with the lowest data delivery delay. VADD assigns a cost to each road (segment between intersections) based on the expected packet forwarding delay ($d_{ij}$) between intersection $I_i$ and $I_j$. If the vehicle density on that road is high (e.g. enough to forward packets in a multihop fashion), $d_{ij}$ is equal to the Euclidean distance of between intersections. If the vehicle density on that road is low, vehicles may carry the data. The authors propose a stochastic model to estimate the data delivery delay ($D_{ij}$). Once calculated $D_{ij}$, and under arriving to an intersection, the vehicle may choose that road with smallest $D_{ij}$. Since on that road and giving the direction of cars several contacts may be chosen as a forwarders, several strategies based on vehicle movement and direction are presented. To calculate vehicle density at each road, the authors assume that vehicles arriving at intersections follow Poisson distribution. The protocol is compared with DSR (topological), GPSR (geographic) and epidemic routing protocols, showing VADD better performance that those protocols.

2.3.6 Mobility Models and Simulators

Mobility models determine the location of nodes in the studied area as a function of time. Mobility Models (MM) in MANETs basically consider synthetic models that aim to represent as realistically as possible mobility patterns without the use of traces. Camp et al [31] discuss several MM in MANETs and classify them as (i) Entity Mobility Models that mimics the mobility of single nodes and (ii) Group Mobility Models that mimic the mobility of groups of nodes. Examples of these models are the following:

- Entity Mobility Models
  - Random Walk MM
  - Random Waypoint MM
  - Random Direction MM
  - A Boundless Simulation Area MM
  - Gauss-Markov MM
  - A Probabilistic Version of the Random Walk MM
  - City Section MM
  - Obstacle MM

- Group Mobility Models
2.3. Delay Tolerant VANETS

- Exponential Correlated Random MM
- Reference Point Group MM and its particular cases (e.g. Column MM, Nomadic Community MM)
- Pursue MM

All these MM consider that the nodes move randomly. The main difference among them is whether the direction, the angle, the destination, the speed or the time are chosen randomly. However in Vehicular Networks, the mobile nodes are cars that follow particular mobility patterns (i.e. roads in a highway scenario and streets in an urban scenario) so their behavior differ a lot and the performance results of protocols evaluations could have a great impact depending on the MM used.

Mobility Models in Transport Systems can be classified in macroscopic and microscopic. Macroscopic MM consider as principal motion constrains streets, buildings, traffic lights, etc and as traffic generation traffic densities, traffic flows, etc. Microscopic MM models each vehicle movement. Its motion constrains are the neighboring cars, pedestrians, driver’s habits, etc and its traffic generation considers inter-distances between vehicles, acceleration/deceleration, braking, etc.

In any case, a realistic mobility framework must consider that the vehicle initial and final destination is not random neither uniformly distributed, the vehicles accelerate/decelerate and follow determined routes with velocity speed limitations and different lines and road categories, traffic density depends on the hour and day (traffic peaks), obstacles (i.e. buildings, trucks, hills, etc) impact both mobility and communications. Harri et al [32] classify vehicular MM as following:

- Synthetic Mobility Models as those based on mathematical models. Its main drawback is to validate the mathematical model with real mobility patterns. See Fiore’s paper [33] for a survey on Synthetic Mobility Models.
  - Stochastic models consider purely random models on a graph (e.g. road topology) in which cars follow casual paths over the graph with random speed. They do not model car-to-car interaction, or intersection modeling. Good to compare with simple models such as Random Waypoint Model.
  - Traffic Stream models looks vehicles as hydrodynamic phenomenons relating velocity \( v(x,t) \) in Km/h, density \( \rho(x,t) \) in vehicles/Km) and flow \( q(x,t) \) in vehicles/h). Since flow is a macroscopic parameter, they do not model microscopic behaviors.
  - Car Following models where each vehicle is modeled according to vehicles ahead and therefore as single nodes (microscopic modeling). They may also characterize other parameters such as drivers attitude or lane-changing. An example of this model is the IDM (Intelligent Driver Model) used in VanetMobiSim.
2.3. Delay Tolerant VANETS

- **Queue models** which models roads as FIFO queues with cars and clients. This category falls in the middle between microscopic and macroscopic models.

- **Behavioral models** where each movement follows behavioral rules. Hard in computation terms.

- Survey-based Mobility Models as those extracted by surveys or statistics obtained during several years. These models are able to simulate social activities such as pedestrian behavior, work-day activities, traveling distance, etc. (e.g. UDel MM, Agenda-based MM)

- Trace-based Mobility Models as those extracted from real traces. From the traces some patterns not directly observed may be extrapolated. The main drawback of this method is that the models are biased by the experimental conditions in which the traces were obtained.

- Traffic-Simulators based Mobility Models as those extracted from a detailed traffic simulator. These models are refinements from Synthetic, Traced-based and Survey-based Mobility Models like SUMO [34].

In networking the mobility of each node impacts performance parameters such as link duration, throughput, packet delivery ratio, etc. Therefore, microscopic models are of more interest with respect macroscopic models in which vehicle flows are modeled.

Classical research in Ad Hoc Networks use several networking simulators. In general these simulators are built on several modules in which the user may define mobility models and the OSI layer based on TCP/IP: physical layer, MAC layer, IP layer, a transport layer and a application layer. As stated by Kurkowski [35], in an analysis of works published in a premiere conference of MANETS, the network simulators most used are:

- *ns* 43.8% [3]

- *GloMoSim* 10 % [36]

- *QualNet* 6.3% [37]

- *OPNET* 6.3% [38]

- *Self-Developed* 27.3%

- *others* 6.3%

A first difficulty on evaluating protocols in Ad Hoc Vehicular networks is the fact that transport simulators are designed for evaluating vehicle car densities, vehicle traffic mobility patterns, etc but are not designed to evaluate accurate networking parameters along the OSI layers. On the other hand, Network simulators include a high number of networking protocols proposed by the networking
community, although none of the networking simulators model vehicular mobility patterns. Some efforts have lately been taken to overcome this problem (see [32] for a complete survey on VANET simulators pros and cons):

- **VanetMobiSim** [39] can generate movement traces in different formats, supporting different simulation/emulation tools for mobile networks (NS2, GloMoSim, QualNet)

- **Straw** [40] (STreetRAndom Waypoint) provides simulation results by using a vehicular mobility model on real US cities. STRAW’s current implementation is written for the highly efficient JiST/SWANS

- **TraNS** [41] (Traffic and Network Simulation Environment) is a simulation environment developed at EPFL that integrates both traffic and network simulators. TraNS obtains mobility patterns from SUMO and feeds the networking simulator ns-2.

After this overview of network simulators and Mobility Model generator tools, and a deeper research about pros and cons offered by other researchers who have used them, we have chosen ns-2 and VanetMobiSim to develop our projects. OPNET also has good reviews, it is said to be easier to learning how it works and start getting results but it is not a free option (It is possible to apply for a free license but there are several requirements). On the other hand, ns is known to have a hard learning curve but it is also a powerful tool, open source, and by far the most used, has a big community of users, an active mailing-list to get help and several protocols implemented. We choose VanetMobiSim, because it is easy to use, it has several mobility models implemented including Intelligent Driver Models, and it also has a good documentation and some validation [42], [43].

### 2.3.7 Delay Tolerant VANET applications

Even that some abstraction layers are used, having a certain general understanding of how networks work is always needed to develop reliable network-based applications. In the case of Delay tolerant networks these understanding requirements are more important. The limitations of these networks are higher and must be taken in consideration when implementing an application.

The most common problems applications found when deployed over delay tolerant networks are:

- **Timeouts**: reply times may be longer than foreseen by timeouts, they are used to reinitiate a transaction or to warn the user that some connection could not be accomplished. So, especial attention must be paid when deciding timeouts, or if possible, avoid them.

- **Synchronous Programming Style**: In common applications, it is assumed that the total execution of an application will be longer than its connection
2.3. Delay Tolerant VANETS

establishments, and both ends will be synchronized or running at the same
time.

- Chatty Application Protocols: with short end-to-end round trip times, you
can afford to have several exchanges to establish a communication.

SOTIS (Self-Organizing Traffic Information System) is a proposal within the
FleetNet project. SOTIS makes the basic assumption that the decision of a driver is
most likely influenced, see [44], by information on its surrounding area. Therefore,
vehicles periodically broadcasts its knowledge about the traffic situation (e.g. its
velocity, emergency information, etc.) to other vehicles within its transmission
range.

Comfort applications require different kind of requirements. For example, in
FleaNet [45], users express their demands/offers via radio queries. These queries
are opportunistically disseminated in order to find other customers. In this pro-
posal, the authors create a market place over vehicular networks and identify key
performance metrics (i.e query resolution latency, scalability and mobility).

The search for free parking places is a problem in big cities. Caliskan et al
mention in [46] a German study on the annual damage resulting from searching
parking space. The study results in an economical damage of 20 million Euros, 3.5
millions Euros for gasoline and diesel, a 150.000 hours of waiting time and 44% of
the total traffic are vehicles looking for free parking places. Papers [46] propose
dissemination algorithms and prediction of parking occupancy using VANETs.

SPAWN, see [47] is a cooperative strategy for content delivery in a VANET.
The work is based in a gossip mechanism to propagate content availability infor-
mation and a proximity driven content selection strategy. Vehicles have reduced
intervals of time to download files in roads with fixed infrastructure separated at
regular points. A car arrives at a gateway and downloads pieces of files (Bit-
Torrent style). After getting out of the gateway range starts a gossip with its
neighbors about content availability and exchange pieces of the file obtained by
neighbor vehicles. Content selection is based on several strategies that reduces
contention.
Chapter 3

Framework

3.1 Scenario description

The intended scenario, see Figure 3.1, consists in a road where vehicles use 802.11-based devices to download contents from road-to-car infrastructure using a mesh of APs. Several APs are uniformly or randomly distributed through the road/highway, but not so close to allow a non-disruptive connection. APs act as custodians in a Delay Tolerant Network (DTN), keeping files for nodes. It is not the purpose of this master thesis to define the DTN architecture. We will assume that there is a server that schedules which files are stored at each AP and which and how packets are sent to the vehicles.

![Figure 3.1: The intended scenario consists on a road with sparse APs](image)

In the considered scenario, vehicles accessing an AP have few seconds to download information and due to harsh conditions the losses produced in such environment are high.

In [1] we performed some experiments with 3 cars in a urban scenario experimenting 20-30% of losses, of course, lower than the reported in [48] because of the speed of vehicular nodes and the lower data rate we employ for the tests. But as can observed on Figure 3.2, different vehicles even traveling closer will have different reception rates, in this picture we can observe the probability of reception in the three cars of packets addressed to the second car. This Figure was taken experimentally with 3 cars at 20 Km/h and a fixed AP (the experiments are explained in section 3.3.3). Take notice that for example when car 2 is entering the coverage area (Region I), the probability of reception at car 1 is better because it is closer to the AP. So in such urban scenarios, and in other faster road scenarios, there are nodes that can cooperate, so a cooperative-ARQs is an efficient and feasible mechanism.
3.2 Simulation

Simulations were performed with the standard version of the ns-2.31 simulator [3]. Each AP generates packets to vehicles in its coverage area in round robin basis. A simple registration and keep alive signal, see Figure 3.3, is used to notice when nodes are in coverage and when they leave an AP.

Vehicles request files of 10 MB and packets have a size of 1 KB. Vehicles travel 30 Km on a two-way highway with two lanes per direction. The road network infrastructure consists of 5 Access Points placed every 6 Km. Vehicles move with constant speed, randomly chosen from a uniform distribution between 70-90 km/h on the right lane and between 90-120 km/h on the left lane. Vehicle density is modeled with exponential distribution of parameter $\lambda_1$ vehicles/s in the
3.2. Simulation

<table>
<thead>
<tr>
<th>Antenna</th>
<th>Omnidirectional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>5.9 GHz</td>
</tr>
<tr>
<td>RxTh</td>
<td>-95 dBm</td>
</tr>
<tr>
<td>CSTh</td>
<td>-96 dBm</td>
</tr>
<tr>
<td>Antenna Gain</td>
<td>1 dB</td>
</tr>
<tr>
<td>TxPower</td>
<td>9.95 dBm</td>
</tr>
</tbody>
</table>

Table 3.1: Configuration parameters

right lane and $\lambda_2$ vehicles/s in the left lane. These rates consider the maximum number of vehicles following vial rules (e.g. security distances of 80 meters traveling at 90 Km/h, 100 meters traveling at 100 Km/h, etc). Using these consideration, $\lambda_1=0.25$ vehicles/s in the right lane and $\lambda_2=0.2$ vehicles/s in the left lane would fill the highway with the maximum number of vehicles traveling at 100 Km/h and at 120 Km/h at each lane. Higher densities may be achieved lowering vehicle speeds (then the security distance between vehicles is lower). In the graphs we normalize $\lambda_i (i = 1, 2)$ and for clarity we define parameter $\alpha$ as vehicle density, being $\alpha = 1$ the higher density corresponding to vehicles with 120 Km/h as maximum speed at line 1 and 90 Km/h as maximum speed at line 2. Lower values of $\alpha$ indicate a decrease in vehicle density.

For the physical layer we consider IEEE802.11a at a rate of 3 Mb/s. Nodes use omni-directional antennas, see Table 3.1.

We have used Nakagami as propagation model. This model is used to predict signal attenuation in fading environments and has already been used in vehicular scenarios, [49]. The Nakagami probability density function defines a distribution of the power $x$ of the received signal:

$$f(x; m, \Omega) = \frac{m^m}{\Gamma(m)\Omega^m} x^{m-1}e^{-\frac{m}{\Omega}x}$$

Where, $\Gamma$ denotes the Gamma Function, $\Omega$ denotes the average received power, $m$ is the Nakagami parameter and both $\Omega$ and $m$ depends on the distance between transmitter and receiver. High values of the m-parameter (with $m > 1$) introduces a variability on the average power reception similar to two ray-model while a value of $m = 1$ introduces Raleigh distribution variability. Lower values of the parameter $m$ worsen channel performance. That allows to define a reception similar to two-ray mode in short distances with reception powers that depends on the $d^{-2}$ and reception similar to fading models in larger distances with reception powers that depends on $d^{-\gamma} (\gamma > 2)$.

Figure 3.4 shows delivery ratio for two path loss scenarios and two antenna gains. For this simulation, nodes send periodic hellos to allow the AP to know they are in coverage. While the nodes are in coverage, the AP sends packets. The delivery ratio showed is the number of packets received per packet sent. Scenario A uses a nakagami that considers a two-ray model in short distances ($d < 200$ meters). At distances higher than 200 meters, the reception power decreases at a
power of $\gamma$. The fading channel is the same at all distances: a high loss fading channel of $m = 0.5$. Scenario B is very similar to scenario A, but with different fading channel depending on the distance. Distances lower than 80 meters have a Raleigh fading channel, from 80 to 200 meters the channel is worst ($m = 0.75$) and finally at higher distances ($d > 200$ meters) $m = 0.5$ (an even worst fading channel). As a summary Scenario A is more pessimistic than Scenario B.

![Figure 3.4: Delivery ratio as a function of car density](image)

Remember that the $\alpha$ depicted on the x-axis means the normalized car density with respect to the maximum possible. It can be observed how delivery ratio slowly decreases when density of nodes increases, due to higher collisions of the hello packets used by the registration mechanism. It can also be observed the impact of the antenna gain in the delivery ratio. As expected, results show a worst delivery ratio on scenario A, we chose this scenario for the simulations.

### 3.3 ARQs for Delay-Tolerant Vehicular networks

#### 3.3.1 End-To-End ARQ

The scenario consists of an end-to-end message-oriented delay tolerant network where the APs, known as custodians, use persistent storage to save the bundles to reach the destination. We will assume that there is a server that schedules which files are stored at each AP and which and how packets are sent to the vehicles.

Due to the mentioned harsh conditions on vehicular networks scenarios, the time nodes spend on coverage areas must not be wasted. Our aim is to optimize the chance of communication to achieve the best performance. An end-to-end
ARQ with several coordination packets would waste too much time. As stated before at 2.2.4 the use of TCP or a TCP based modification would severely limit the throughput.

Due to this problems the network would need too long to start a transmission, or would use bandwidth in retransmissions for nodes that are already out of coverage, and losses due to the wireless physical medium mess up TCP because packets loss are considered to be the result of congestion, then the congestion window size is reduced and after an erroneous back-off of the congestion window, there would be a congestion avoidance phase with a conservative decrease in window size.

To combat these harmful effects, we apply end-to-end modifications at the client and at the server, and modifications in the network layer, to apply custodian based solutions. We analyse how ARQ mechanisms with lower coordination behave and how they improve global and local performance parameters.

In our scenario we will suppose that, even we may have sporadic end-to-end connection, it will not be enough time to download the entire bundle or file. And destination nodes (vehicles) will use several custodians (APs) to download a file. Therefore, the network is formed by the sender, the receiver and some custodians. We suppose that in our Vehicular frame, the sender and the custodians have a high bandwidth wired connection. Allowing them a reliable communication. So, even that the ARQ communication takes place between the destination and a custodian, the other custodians could be aware about the end-to-end ARQ status when the node leaves the current custodian, in order to the next custodian could avoid retransmitting packets which the destination has already acknowledged.

With regard to the identification and association phase, vehicles detect AP along the road and register its willingness to download packets. Suitable mechanisms for detecting in-range AP, association and authentication of vehicles reaching a given AP must be provided. Those mechanisms can have a major importance on the overall performance, but are not related to the ARQ, and thus we leave them out of the scope of this Master Thesis. It can be assumed, for example, that vehicles are equipped with WAVE IEEE 802.11p cards. WAVE architecture provides mechanisms to access WAVE Base Stations (AP) in vehicular networks. In our framework, AP periodically broadcast hello beacons each 0.2 seconds, identifying their zone. Each node that receives the beacon knows it is in coverage and will reply with a registration packet containing its identification.

Files (or bundles) of size $S_F$ are divided in blocks of size $S_B$ packets, so there are $\frac{S_F}{S_B}$ blocks. ACK packets include a bitmap of a fixed size, which acknowledges packets received by the destination. As it would be excessively long to acknowledge the entire file, it only acknowledges some blocks. In addition to this bitmap, it is also piggybacked another smaller bitmap that summarizes the previous bitmap, acknowledging completed blocks.

So, on one side we have the sender of the ARQ packets, this would be the node which is receiving a string of bytes from a bundle and it should acknowledge/unacknowledge the correctly received/unreceived ones. At the other side, the receiver of the ARQ packets, it is the custodian that is sending the bundle,
3.3. ARQs for Delay-Tolerant Vehicular networks

and should retransmit the non received packets.

To study the ARQ we have proposed 3 mechanisms for the sender of the ARQ packets, and 3 more for the receiver of the ARQ packets, and we study the behavior of their combinations.

Sending acknowledges behaviors:

1. First Block (FB): starting from the first block not completed

2. Round Robin Block (RRB): starting from the begin of a not completed block selected applying round robin


Receiving acknowledges behaviors:

1. Retransmissions First (ReTxF): sends the nack packets first, and continue

2. Rounds of Retransmissions (RReTx): transmits the file once, and then complete rounds of retransmissions

3. One Round in each AP (R1AP): only one round to the file in each AP

ARQ Mechanism from the sender’s side

From the sender’s side, when it needs to send an ACK it could use:

- **First Block (FB):**

  See Figure 3.5, the sender of the ARQ packet acknowledges packets starting from the first block of the bundle which is not completed, and filling the length of the bitmap with the following blocks. This method gives higher priority to the first blocks of the bundle, so the receiver will have a detailed and fresh view of them, in exchange of no information about the latter blocks is received by the destination.

![]()  

**Figure 3.5:** First Block (FB) Mechanism
3.3. ARQs for Delay-Tolerant Vehicular networks

- **Variation First Block-Tetris (FB-T):**
  Variation ("Tetris"): See Figure 3.6, if a block is completed it "disappears" from the packet’s bitmap. The destination can know which blocks are completed using the blocks’ bitmap, so if one of the blocks is completed we remove or cut their bits in the packet’s bitmap, including the next, allowing the ACK to be more useful, because it does not contain redundant information that could be summarized in the blocks’ bitmap.

The block’s size could have an important impact on the performance. Using big blocks we obtain a smaller bitmap, but it would be more difficult to complete a block. Therefore, some packets of nearly completed blocks would be acknowledged several times and information about unreceived packets would be small when blocks are nearly complete. On the other hand, in case of using smaller blocks we need bigger blocks’ bitmap, but it would be easier to complete a block and summarize it, therefore the information at the packet’s bitmap would be more useful because it would contain more information about unreceived packets.

![Figure 3.6: First Block Tetris (FB-T) Mechanism](image)

- **Round Robin Block (RRB):**
  See Figure 3.7, the sender of the ARQ packet acknowledges packets starting from the first packet of a not completed block, selected applying round robin, and filling the length of the bitmap with the following blocks. This method gives higher priority to having a wide knowledge of the packets received from all the in course blocks. Understanding by "In course blocks" the blocks that have at least one packet transmitted.

![Figure 3.7: Round Robin Block (RRB) Mechanism](image)
3.3. ARQs for Delay-Tolerant Vehicular networks

- **Variation Round Robin Block-Tetris (RRB-T):**
  See Figure 3.8, the same method explained in Variation First Block-Tetris is applied to Round Robin Block.

![Figure 3.8: Round Robin Block Tetris (RRB-T) Mechanism](image)

- **Complete Bitmap (CB):**
  See Figure 3.9, this ideal case, supposes it is possible to acknowledge the whole file. It is used to know how well the other mechanisms behave in comparison with an ideal case where it were possible to (un)acknowledge the whole file. In this case a "tetris" variation like the others is useless, because you already have all the information.

![Figure 3.9: Complete Bitmap (CB) Mechanism](image)

**ARQ Mechanism from the receiver’s side**

From the receiver’s side, when it receives an ACK it could use:

- **Retransmissions First (ReTxF):**
  When an ACK is received, if there is any packet unacknowledged, the node retransmits all of them. After finishing, the transmission returns to the place where it was when it was interrupted by the ACK. Special attention must be paid to differentiate packets lost from packets which are not sent yet, or are sent short before and may be "on the air".

- **Rounds of Retransmissions (RReTx):**
  When an ACK is received, if there is any packet unacknowledged, the node takes notice of it, but it continues transmitting the packets of the file sequentially. After finishing the first round to the file, a new round of retransmissions starts, and new rounds of retransmissions will be done until the destination completes the bundle or it runs out of coverage.
3.3. ARQs for Delay-Tolerant Vehicular networks

- **One Round in each AP (R1AP):**
  This works like the mechanism before, what changes is that only one round is done in each AP. What you learn from the ACK will be used by the next AP the node will visit, which will have a precise knowledge of the packets received and no useless retransmissions are done. So, in comparison with the other mechanisms, less retransmissions are expected but higher delays to download files.

  The intended behavior of the end-to-end ARQ could be seen in Figure 3.10, download start in one AP and if it is not ended, it could continue with the following, retransmitting packets lost and new blocks of the file.

![Figure 3.10: Expected behavior of delivered packets using an end-to-end ARQ](image)

### 3.3.2 Delayed Cooperative ARQ

Although and end-to-end ARQ mechanism is needed to recover packets, we may also use the broadcast nature of the media to try to recover packets. Cooperative techniques may improve performance in terms of packet losses. When a vehicle receives a packet, its neighbor nodes may independently receive the packet. In this way, some of the vehicles in the platoon traveling with the destination node may have some of the lost packets. Nodes, then, may cooperate to recover packet losses minimizing the number of end-to-end retransmissions.

Consider Figure 3.11 in which vehicles want to download information from the Internet through AP distributed along a road. The key idea of this scheme is to delay retransmissions coming from cooperators until the platoon is on a dark area, out of coverage of any AP. This scheme operates into three phases:

1. **Association phase:**
   Vehicles detect AP along the road and register its willingness to download packets. This phase still works like the explained in section 3.3.1. AP periodically broadcast hello beacons each 0.2 seconds, identifying their zone. Each node that receives the beacon knows it is in coverage and will reply with a registration packet containing its identification.
3.3. ARQs for Delay-Tolerant Vehicular networks

2. **Reception phase:**

Vehicular nodes receive data from the AP. Each car receives its data but also buffers packets addressed to other cars in the platoon that it considers as co-operators. The cooperation relationship is established through the exchange of HELLO messages broadcasted periodically by the vehicular nodes. The first function of a HELLO message sent by a node \( x \) is to allow other nodes to know about the presence of \( x \). Cooperators do not retransmit the packets while they are in Reception phase. Retransmission (i.e. cooperation) of packets will be delayed until the platoon of cars is out of the coverage area in a new phase that we call Cooperative-ARQ phase. In this phase data flow is always from the AP to the vehicular nodes, and no retransmissions are used, unless the current end-to-end ARQ decides to do it, we avoid retransmissions at the hope that cooperators will receive packets incorrectly received by the destination and will help it in the Cooperative-ARQ phase, without the need of wasting the useful time in coverage with the AP in retransmissions. In this way the channel can be used by the AP to transmit as much new data addressed to the cars as possible, thus reducing the downloading time and increasing the effective data rate.

3. **Cooperative-ARQ phase**

When the cars leave the AP range, they enter into the Cooperative-ARQ phase. After some time (e.g. equivalent to 10 beacons) without receiving beacons from the AP, a node considers that is out of the AP coverage. At this point, every node checks which packets it has failed to receive correctly from the AP and starts to request them to other vehicular nodes (i.e. to its cooperators), in an attempt to recover all packets from the first to the last received from the AP. The process is the following: (i) A node \( x \) broadcasts a REQUEST packet for each started block that it has failed to complete from the AP with its packet's received bitmap. (ii) When receiving this REQUEST, each cooperator of \( x \) will check if it has any packet from the requested block buffered (it has received the packet correctly from the AP in the previous phase). (iii) If it has some packets, it will send the packets to \( x \) (unless other cooperator sends it before). This process will be repeated while
the node receives any packet from its requests, and ends after the tenth time that any cooperator haven’t replied or when it enters in range of a new AP, meaning that it comes into reception mode (Reception phase of the protocol operation), and the whole cycle starts again. Note that the end-to-end block ARQ still is working on top of the cooperative ARQ mechanisms, packets not recovered from cooperation will be retransmitted by the end-to-end ARQ.

The intended behavior of the end-to-end ARQ with the DC-ARQ could be seen in dark gray at Figure 3.12, light gray is the intended behavior without DC-ARQ, download start in one AP and if it is not ended, it could continue with the followings. In addition, once a node has left the AP coverage, it could receive cooperation from cars of the same platoon, and when arriving to the next AP the end-to-end ARQ would continue retransmitting packets not recovered and transmitting new blocks of the file.

Figure 3.12: Expected behavior of delivered packets using an end-to-end ARQ and DC-ARQ

3.3.3 Experimental Prototype validation

To check the feasibility of this mechanism, we performed a set of experiments using a prototype developed by Julian Morillo in his Thesis. The results were published in [1], the main outcome of the experiments are: (i) That the proposed protocol can effectively reduce the packet losses of transmissions from access points to cars in a platoon, and (ii) as a side result, we found that the relative position of the cars can have a great impact on the opportunities for cooperation (i.e. if two cars are close enough while receiving data, they will have similar reception conditions, thus reducing the opportunities for cooperation). In any case, we have demonstrated, with a prototype of the mechanism, that an almost optimal performance can be achieved in the sense that, given the packet receptions on each car in the platoon, each car is able to recover all the packets it has lost from the access point from the other cars provided that they have them. The experimental results give promising improvements to take into account for future research on the field.
3.3. ARQs for Delay-Tolerant Vehicular networks

Note that an important conclusion that can be derived from these results is that the mechanism can effectively reduce the number of access points that a car needs to access for, for example, downloading a certain file. Moreover, this loss reduction can be used also to increase the transmission rate of the APs, while keeping a reasonably loss rate.

Experimental Setting

The tests were performed in the urban scenario depicted in Figure 3.13. The AP was located in the position marked as AP in Figure 3.13 and consisted in a desktop PC equipped with a Proxim external PCI wireless antenna located in an office in the first floor of the building. The antenna was located on the window of this office.

![Figure 3.13: Map of testbed (Campus Nord, UPC)](image_url)

On the other hand, three mobile vehicular nodes were used, consisting of three laptops (Toshiba Satellite Pro A120 model) equipped with Cisco Aironet AIR-CB21AG-E-K9 802.11a/b/g PCMCIA wireless adapters, each of them transported by a car. The three cars followed the path marked with the white arrows in Figure 3.13, all together at an average speed of about 20 Km/h. for a total number of 30 rounds. We have named them as Car 1 (the first), Car 2 (the car in the middle) and Car 3 (the last).

The implementation of the mechanism was done by Julian Morillo, using Click Modular Router [50] and all the cards were controlled by a MadWiFi [51] driver in monitor mode and with retransmissions disabled. All transmissions (i.e. AP to Car, Car to Car) were performed using 802.11g at 1Mbps.

The AP transmitted constantly three different data flows addressed to each car on the experiment consisting of 5 ICMP Echo Request messages per second with an ICMP payload of 1000 bytes, each one. During the experiments we captured all the received traffic on each laptop for its analysis and postprocessing.
3.3. ARQs for Delay-Tolerant Vehicular networks

<table>
<thead>
<tr>
<th>Car</th>
<th>Tx by the AP</th>
<th>Lost before coop.</th>
<th>Lost after coop.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>130.4</td>
<td>30.5 (23.4%)</td>
<td>13.7 (10.5%)</td>
</tr>
<tr>
<td>2</td>
<td>143.0</td>
<td>38.4 (26.9%)</td>
<td>24.8 (17.3%)</td>
</tr>
<tr>
<td>3</td>
<td>121.4</td>
<td>34.7 (28.6%)</td>
<td>19.1 (15.7%)</td>
</tr>
</tbody>
</table>

*Table 3.2: Average values on the number of packets sent, received and lost in the three cars*

We have performed the tests on an urban scenario for its easiness of deployment in contrast to a highway scenario. However, this simple scenario allows us to show how cooperative techniques and more precisely, Delayed Cooperative ARQ can help on the improvement of these kinds of networks.

**Experimental Results**

Firstly, we present in 3.2 the average values on packet losses obtained along the 30 rounds performed on the experiment. Together with the mean absolute values, we show the percentages of losses without and with the cooperative ARQ mechanism.

As can be seen in Table 3.2, all three cars present an improvement on the reliability on the link between the AP and themselves. Note how car 1 reduces its packet losses from a 23% to a 10%; car 2 from 27% to 17%; and car 3 from 29% to 16%. Especially striking is the case of car 1, where a reduction of more than 50% in the number of lost packets is achieved. It seems strange that the poorer results are obtained for car 2. This fact, however, can be explained, as we will see, by the environment and by the different behavior of the three cars along the experiment (distances between them, etc.). In other tests, not shown here, car 2 normally achieved the best performance. This is a normal result as car 2 is the car located in the middle of the platoon and can benefit from the cooperation coming from car 1 on the first range of packets it should have received from the access point while entering the coverage area and from the cooperation of car 3 on the last packets that it should have received while leaving the coverage area.

Now that we have seen the mean values obtained on the experiment, let us focus into the details to explain them and study the probabilities of reception of packets on the different cars.

In Figure 3.14 the probability of reception for the three different cars of packets addressed to car 2 is shown. Three different packet reception regions can be defined: Region I corresponds to car 2 entering the AP coverage area while car 1 is on the coverage area and car 3 is outside. Region II corresponds to the car 2 on the coverage area, while car 1 is starting to leave and car 3 is entering. In Region III, car 2 is leaving the coverage area while car 3 is still there.

We can observe on Region I (first packets) that, as expected, car 1 has better reception conditions (as long as while car 2 is entering the coverage area, car 1 is expected to be in the center of it), so car 2 will benefit from cooperation coming from car 1.
3.3. ARQs for Delay-Tolerant Vehicular networks

When car 2 starts to leave the coverage area (Region III), a better reception probability on car 3 was expected. It is also important to note that, while on Region I of the figure, car 2 and car 3 perform quite different, on Region III their probabilities of reception are almost the same. We argue that this is because of the behavior of the different drivers. The fact is that the driver in car 2 was the least experienced, thus meaning that at corner marked as C on Figure 3.13, car 3 became very close to car 2 in almost all rounds, making their reception conditions on the street before the corner (after turning to the right on corner C) quite similar.

The next figure 3.15 present interesting results taking a different approach. In them the probability of correct reception using C-ARQ (i.e. after cooperation) is compared with the joint probability of reception of the different packets in car 1, 2
3.3 ARQs for Delay-Tolerant Vehicular networks

or 3 in order to establish if the C-ARQ mechanism implemented and tested works properly and the effectiveness of it. The two curves are almost coincident indicating that the protocol works almost optimally in the sense that the destination car is able to recover all the packets that have been received in any of the cars in the platoon. Here is the key idea behind the mechanism: it exploits the diversity that can be achieved thanks to the different cars on a platoon and performs as well as a virtual car which uses the better reception conditions of all of them.

3.3.4 End-to-End ARQ and DC-ARQ performance evaluation

For the evaluation of the mechanisms we have simulated all the combinations that have sense. Avoiding for example the use of First Block (FB) with Rounds of Retransmissions (RReTx), in this case, if APs retransmit all the file but the node only acknowledges the first blocks, the APs have no knowledge about end blocks possibly completed and may be retransmitting them uselessly.

Figure 3.16: Average # of Rtx to Download a 10MB file, (10,000 pkts). Shows all the proposed ARQs, The better are shown more clearly at next figure.

Figure 3.16 shows the average number of packets retransmitted to download a 10MB file, (10,000 pkts). The total number of transmissions would be these retransmissions plus the fixed 10,000 packets. First block with Retransmissions First is far from the performance of the rest, but when used with Tetris variation its performance improves and becomes more similar to the others. This happens because:

1. the ACK may have some packets lost at block $K_i$ and the following $K_{i+1}$ could be a completed block also included in the bitmap, so few information about the following blocks is useful for the AP.
3.3. ARQs for Delay-Tolerant Vehicular networks

2. The AP retransmits the few unacknowledged packets and continues from the last not acknowledged block.

3. The next ACK is sent and in the worst case a packet may be still lost at the first $K_i$ block.

4. The AP retransmits the packet and continue from the last not acknowledged block, which are the same as before, these not acknowledged blocks are being retransmitted uselessly because we do not know which packets are already received, and it increases a lot the number of retransmissions.

Figure 3.17: Average # of Rtx to Download a 10MB file, (10,000 pkts). Shows only the best performing ARQs

Figure 3.17 is the same as Figure 3.16 but without the $FB \text{ ReTxF}$ combination, to see more clearly the differences between the rest. There is also the best or ideal minimum number of retransmissions with packet delivery ratio of 0.75\(^1\) and 0.7. It shows that ReTxFirst introduces the highest number of retransmissions combined with all the sender’s ACK mechanisms. As expected, the ideal Complete Bitmap mechanisms perform better. And among the RoundRobin Block Ack, the Tetris variation reduces the number of retransmissions.

Figure 3.18 show the average number of APs used to download the file, and the time they spend since the start of the download. As time between APs is longer and dominates the total download time more than if the download ends at the beginning of an AP coverage or at the end. Delay and # of APs always have the same behavior. In the next evaluations we will show only the average number of APs. Because It is more general, using it and a different distance between APs, it is possible to know the Delay in this different scenario.

\(^1\)The average number of retransmissions with a loss ratio of $r$ could be described by a geometric progression $a/1-r$ minus $a$, where $a$ is the number of packets to be delivered. For $a = 10,000$ packets and a delivery ratio of 0.75($r = 0.25$) the average number of retransmissions is $= 3.334$
3.3. ARQs for Delay-Tolerant Vehicular networks

Figure 3.18 shows that FB ReTxFirst, and as expected, the combinations using One Round in each AP mechanism, will need more APs and also more time, to download the complete file. In the case of FB ReTxFirst this is due to the bad behavior already commented, in the case of the combinations using One Round in each AP mechanism, it is due to the decision of only do one round of transmissions to the file being sent in each AP.

Figure 3.19 is the same as the first plot of Figure 3.18 but only shows the best combinations. The combinations that use Round Robin Retransmissions perform better than the others when the number of nodes downloading increases. The ideal CB with RoundReTx is the best, followed by the RRB-T mechanism. The differences are not high because if few packets, or even one, remain to be received you need to wait until the next AP.
3.3. ARQs for Delay-Tolerant Vehicular networks

Figure 3.19: Average # of APs used to Download a 10MB file. Shows only the best performing ARQs

Figure 3.20: Average # of Rtx to Download a 10MB file (10,000 pkts), changing the Beacon interval

To notice how the period of the beacon impacts the performance, there are Figure 3.20 and Figure 3.21. They show the performance of the ARQs as a function of the periodicity of the beacons. Figure 3.20 shows how with a short time between broadcasts there are higher retransmissions due to higher collisions and because nodes will detect they are in coverage and start the download earlier, from a further distance and so with more time of coverage but a lower average delivery ratio because the extra time is at the ends of the range of coverage, where there are high losses. But on the other hand, using a time between broadcasts too wide may cause a desynchronization at the coordination of the AP, implying the retransmission of packets that the destination has received but has not acknowledged yet. Complete Bitmap mechanisms decrease even when the others stop doing, because with only one acknowledgment it has enough to do not retransmit duplicated packets while the others when working with higher times between acknowledgments may start
3.3. ARQs for Delay-Tolerant Vehicular networks

Figure 3.21: Average # of APs used to Download a 10MB file changing the Beacon interval

doing it. Figure 3.21 shows that although the retransmissions decrease with longer periods the used APs increase because nodes lose more time to notice the presence of the AP and the AP may considered earlier to have lost the coverage. We must notice that in this simulation the network is not at its maximum bandwidth capacity, otherwise a high number of retransmissions would have a major impact on the # of APs used. Sending an acknowledgment each 0.2 seconds is a good trade-off between the retransmissions and the delay to obtain the file.

Figure 3.22: Average Delay to Download a 10MB file changing the normalized car density

Figure 3.22 and Figure 3.23 show the performance of the ARQs as a function of the normalized car density, for the RRB-T and RReTx combination mechanism. Figure 3.22 shows how for a low car density the difference between them is small even with 0.05s and 0.1s but it increases with a higher density because there
3.3. ARQs for Delay-Tolerant Vehicular networks

Figure 3.23: Average Delay to Download a 10MB file changing the normalized car density

will be higher collisions. But for Beacon periods bigger than the 0.1 seconds the number of retransmissions is similar, the beacons do not interfere so much with the download. Figure 3.23 shows how, as stated before, with a higher time between beacons the nodes have less time in coverage and then they need more APs to complete the download. With a low car density short periods perform well but as density increases the average number of APs needed increases beyond the 0.2s performance.

So a conclusion is that a dynamic beacon’s period as a function of the car density would be a good solution and we leave it for future work.

Figure 3.24: Average # of Cooperations Downloading a 10MB file

Figure 3.24, Figure 3.25 and Figure 3.26 show how the best combinations of end-to-end ARQ mechanisms perform when used in combination with the DC-
3.3. ARQs for Delay-Tolerant Vehicular networks

ARQ, as a function of the normalized car density.

Figure 3.24 shows that ReTxFirst reduces to less than half the cooperation opportunities, because the end-to-end ARQ produces more retransmissions at the AP coverage than the others, then there will be less cooperation opportunities. The number of cooperations using Round Robin ReTx combined with Complete Bitmap and Round Robin Block are similar, showing that the behavior of the number of cooperations changes with a different retransmission mechanism used by the AP but it is not so dependent on the acknowledgment mechanism used by the vehicle. As 1-RoundAP-ReTx does not retransmit the same packet twice in the same AP it has the maximum number of cooperations.

Figure 3.25 shows that Round Robin ReTx with Complete Bitmap performs better, followed by RoundRobin Block. And Figure 3.26 shows how using RRB-T 1RoundAP-ReTx, at first even a small increase in the car density implying the
3.3. ARQs for Delay-Tolerant Vehicular networks

possibility of a few cooperations impact the required APs to download.

Figure 3.27, Figure 3.28 show the improvement obtained when using DC-ARQ for two combinations CB RoundRetx and RRB-T RoundRetx.

Figure 3.27: Average # of APs used to Download a 10MB file with/without cooperation

Figure 3.28: Average # of Rtx to Download a 10MB file, (10,000 pkts) with/without cooperation
3.3. ARQs for Delay-Tolerant Vehicular networks

3.3.5 Conclusions

As we have seen, the end-to-end ARQ has an impact in the overall performance, but not only the mechanism used from the sender of the ACK to decide which blocks acknowledge, and the mechanism of the AP to decide in which order retransmit which packets are important. It is also very important the periodicity of the beacons to tune their behaviors.

In this first part we concluded that a mechanism that achieves to behave like ideal case Complete Bitmap would improve a lot the performance. Achieving a low number of useless retransmissions. But by now, without introducing more complex techniques it is not possible to transmit the Complete Bitmap without using several acknowledgment packets that would increase in excess the overhead.

First Block only performs well when applied with the variation Tetris and Retransmissions First. It could be interesting for application that prefere data arriving ordered (Ordered in a block granularity, not ordered packet by packet) than achieving a better throughput.

Round Robin Block performs better with variation Tetris and have not sense with Retransmissions First because APs retransmit all the file but the node only acknowledges the first blocks.

When combined with Rounds of retransmissions it is the second best in the average number of APs used.

Combined with One Round in each AP it uses a number of APs far beyond them, but on the other hand this second combination offers less retransmissions, meaning that in a denser scenario it would offer a better throughput.

The beacons period that performs better is 0.2 seconds, it is a trade off between the time wasted before the node realizes it is in coverage and the collisions. A faster beacon works better in a low car density scenario while a slower beacon reduces collisions in a high car density scenario, so a dynamic beacon’s period as a function of the car density would be a good solution, we leave it for future work.

We have seen that DC-ARQ used with RRB-T Rounds of ReTx reduces the number of APs used. One of the reasons the improvement is not even better is related with what is known as Coupon collector’s problem. The main idea is that it takes very little time to collect the first few ”coupons”. But on the other hand, it takes a long time to collect the last few coupons. Easily the cooperators would have some packets you need, but it may be difficult they have all to complete and in addition if after cooperation only one packet remains the node needs to wait until the next AP.

To sum up, cooperations are done for free at the dark areas where there are not AP coverage, avoiding to interfere with AP coverage zones, where the bandwidth is a very scarce resource. We have shown that each cooperation done means more than one retransmission avoided. At the denser scenario of CB RoundReTx there are approx. 1600 cooperations and retransmissions are reduced more than 2100 packets, this difference is even greater for RRB-T Rounds of ReTx. Reducing the retransmissions mean that the overall throughput of the network will improve.
3.4 Delay Tolerant Vehicular Protocol

We maintain the same scenario detailed at 3.1. As explained, this infrastructure does not cover the whole path followed by the vehicle, in the figures non coverage zones and coverage zones have a size of the same order, but in reality non AP-coverage zones may be comparatively larger (e.g. 1AP each 6 – 10Km). So coverage opportunities are a scarce resource that when the opportunity appears must be well-spent, this is the aim of DC-ARQ.

DTVP includes this DC-ARQ, but in addition it adds a new type of cooperation, nodes that cross an AP with no download intention, could use the opportunity to cooperate when a downloading node travels in the opposite direction and they will meet in a dark (non AP-coverage) area, so it could carry and forward some packets for this destination.

To allow this, each time a vehicle reaches coverage of an AP (i.e. $AP_k$), registers its direction, average velocity and identity in a server. We assume that the first time a vehicle enters the network, obtains identification (Node-ID). It is not the purpose of this work to define a naming or identification mechanism, neither to define the backbone network.

Access points have storing capabilities and are able to store a whole file or blocks of files. The server schedules which AP stores files or blocks of files that have to be transferred to vehicles. Simple scheduling decisions can be taken from the registered information delivered by vehicles each time they enter AP coverage. The server transfers packets to AP custodians using TCP/IP stack.

AP divides files in blocks of size $L_B$, having thus $B = N/L_B$ blocks per file, where $N$ is the size in byte of the file. Each block will contain $P$ packets of size $L_p$, the L2 MTU (Maximum Transfer Unit). Vehicles that want to download a file send a query to the server. From the vehicle registered information, the server predicts to which AP has to send the file. We may conclude two vehicle location situations: (i) a vehicle is in AP coverage, or (ii) a vehicle has left $AP_k$ coverage area and is driving from the last registered direction at average velocity $v_k$ (km/h). Without any optimized scheduling discipline defined in the server, next time the vehicle enters $AP_{k+1}$ coverage, it registers, updates its location and velocity information and continues downloading packets. The vehicle lasts an average of $T_k = d_k/v_k$ seconds to cross $AP_k$ coverage area (where $d_k$ is the AP coverage in km) and is able to download $N_k = T_k \cdot V t_k$ bytes assuming an average throughput of $V t_k$ bytes/s when traversing $AP_k$. If the file size is $N$ bytes, the vehicle will need ”A” access points where ”A” is that $k$ index that makes $N_1 + ... + N_A = N$. Furthermore the time $T$ needed to download the file is:

$$T \leq \sum_{k=1}^{A} T_k + \sum_{k=1}^{A-1} \frac{D_k}{v_k} + \frac{D_0}{v_0}$$  \hspace{1cm} (3.2)

where we have considered without any loss of generality that the vehicle travels at average velocity $v_k$ from $AP_k$ to $AP_{k+1}$ and the distance from $AP_k$ to $AP_{k+1}$ is $D_k$ km. $D_0/v_0$ is the time lasted from the initial query was issued to the registration in
3.4. Delay Tolerant Vehicular Protocol

the first AP and the reason of the inequality is due to the fact that in the last AP, the vehicle may finish downloading the file before leaving AP coverage. We should firstly note that the $D_k/v_k$ are the terms which dominate transfer delay (i.e. time spent traveling from one access point to the next one). Therefore, any technique aimed to (i) maximize transfer opportunities when vehicles cross AP coverage and (ii) use of gap areas (i.e. time spent traveling between AP) will improve global transfer delays.

DTVP reduces packet losses and optimizes global transfer delay using the following mechanisms: (i) an end-to-end ARQ to provide reliability; (ii) a Delay Cooperative ARQ (DC-ARQ) scheme with neighbor vehicles to minimize packet losses and (iii) a carry and forward scheme in vehicles traveling in opposite direction that opportunistically will cross with the vehicle in the road to improve total transfer delay.

DTVP needs an end-to-end ARQ mechanism that operates at transfer opportunities. In 3.3.1 we defined an end-to-end ARQ mechanism between vehicles and custodians looking also for a trade off between the end-to-end ARQ mechanism and DC-ARQ. In 3.3.2 we showed that packet losses could drastically be reduced using a DC-ARQ mechanism, improving the throughput obtained by vehicles when downloading information from an AP. We integrate them in the DTVP framework.

3.4.1 Carry-and-forward in opposite direction

The server knows from the registered information which AP will be visited by each vehicle. In a highway/road scenario, vehicles follow the road for long distances. In case there is a cross-point, the number of possible next AP that the vehicle will encounter is reduced. The server uploads the file to future possible APs. These APs will be the next information custodians. In case a vehicle does not cross the AP in a reasonable amount of time, the AP removes the file. APs may also delete files directed to vehicles that have already left AP coverage. It can be assumed that there will be vehicles with wireless interfaces that are not downloading information. These vehicles may cooperate carrying and forwarding packets directed to vehicles that they would encounter. The concept is related to epidemic routing [23], and Prophet protocol [28]. In highway/road scenarios, vehicles traveling in opposite direction will encounter each other with high probability. Vehicles with wireless capabilities also register location and direction at custodians. A scheduler defines a delivery discipline to these DTVP-Cooperators in which blocks of packets are selected to be carried-and-forwarded to other vehicles. Let $S_k(t)$ be the number of vehicles crossing $AP_k$ coverage area at time $t$. $S_k$ is obtained from periodic beacons sent by vehicles and can be expressed as $S_k(t) = S_k^→(t) + S_k^←(t)$, where the arrows indicate the vehicles traveling directions. Thus, $AP_k$ serves packets to these $S_k(t)$ vehicles. Furthermore, $AP_k$ receives notification from $AP_{k-1}$ (respectively from $AP_{k+1}$) on vehicles that will arrive to its coverage. Let us call $M_k^{→-1}(t)$ the number of vehicles traveling from $AP_{k-1}$ towards $AP_k$ and $M_k^{←+1}(t)$ the number of vehicles traveling from $AP_{k+1}$ towards $AP_k$ at time $t$. Therefore, $AP_k$ is aware of $K(t)$
vehicles: those under its coverage area plus those traveling towards its coverage area.

\[
K(t) = S_k^-(t) + S_k^+(t) + M_{k-1}^- (t) + M_{k+1}^+ (t)
\]

(3.3)

For simplicity, we consider a lineal road and leave for further study the general case in which the road has bifurcations. Not all \( S_k(t) \) have to be downloading files from \( AP_k \). Let be \( P_k^- (t) \) (respectively \( P_k^+ (t) \) in the opposite direction) the number of vehicles downloading packets from \( AP_k \) and \( Q_k^- (t) = S_k^-(t) - P_k^- (t) \geq 0 \) (respectively \( Q_k^+ (t) \)) the number of vehicles that cross \( AP_k \) and are not downloading packets from the \( AP \). Clearly, \( Q_k^- (t) \) (respectively \( Q_k^+ (t) \)) are candidates to carry-and-forward packets to those vehicles traveling towards \( AP_k \).

DTVP scheduler has to choose (i) how many and which vehicles (called DTVP-Cooperators) among \( Q_k^- (t) \) (respectively \( Q_k^+ (t) \)) will carry-and-forward packets for other vehicles (called DTVP-Receivers), (ii) for what DTVP-Receivers, the DTVP-Cooperators will carry-and-forward packets, and finally (iii) which packets or blocks have them to carry for each DTVP-Receiver.

1. **Election of DTVP-Cooperators:**
   If there are no nodes downloading from an AP the DTVP chooses a DTVP-cooperator. If it is possible it is elected applying Round Robin among nodes nearer than a distance threshold \( d \), choosing for \( d \) a value that ensure nodes good delivery ratios. Otherwise, if there are not enough nodes closer than this distance, all the nodes in coverage could be elected as DTVP-cooperators.

2. **Election of DTVP-Receivers:**
   DTVP-Receivers among \( M_{k-1}^- (t) \) (respectively \( M_{k+1}^+ (t) \)) are elected applying also Round Robin among the \( M_{k-1}^- (t) \) nodes that have just left the \( AP_{k-1} \), a node could be elected for more than one cooperation.

3. **Data to be delivered to each DTVP-Cooperator:**
   So if it has been possible assign a DTVP-cooperator and a DTVP-Receiver, then the scheduler delivers \( B \) blocks to the DTVP-cooperator addressed to the DTVP-Receiver. These blocks are chosen starting by the one with more packets not sent, or unacknowledged. When DTVP finishes to transmit the unsent or unacknowledged packets of the \( B \) blocks, DTVP starts again electing the next DTVP-Receiver and the next DTVP-cooperator. The number of blocks \( B \) depends on the contact time that a DTVP-Cooperator will have with a DTVP-Receiver when they cross and the fairness of the system.

The intended behavior of the DTVP framework could be seen in dark gray at Figure 3.29, light gray is the intended behavior of the previous frameworks with and without DC-ARQ. Download start in one AP and if it is not ended, it could continue with the followings. In addition, when a node leaves the AP coverage, using DC-ARQ and DTVP Cooperators it could receive cooperation from cars of the same platoon, and from cars going in the opposite direction.
3.4. Delay Tolerant Vehicular Protocol

![Figure 3.29: Expected behavior of delivered packets using an end-to-end ARQ and DC-ARQ](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>Normalized car density</th>
<th>average cars/Km</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene-A</td>
<td>1/16</td>
<td>2.5</td>
</tr>
<tr>
<td>Scene-B</td>
<td>1/8</td>
<td>5</td>
</tr>
<tr>
<td>Scene-C</td>
<td>1/4</td>
<td>10</td>
</tr>
<tr>
<td>Scene-D</td>
<td>1/2</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 3.3: Scene Densities, normalized car density with respect to the maximum possible, see section 3.2

3.4.2 DTVP performance evaluation

For the evaluation of the DTVP we have used the end-to-end ARQ combination that performed better after the ideal case Complete Bitmap, this is Round Robin Block with Rounds of Retransmissions.

We will see the averages of the number of APs used, DC-ARQ Cooperated packets and DTVP Cooperated packets as a function of $\delta$, the normalized number of cars downloading. $\delta = 1$ means that all the cars are always downloading. A $\delta < 1$ means that $\delta$ of the cars are downloading at any moment. When one ends, randomly one of the cars not downloading or the same car that have just ended, start a new download. So, $\delta$ is a constant value in time.

By DTVP Cooperated packets we mean the ones received using carry-and-forward in opposite direction.

To see the behavior of the DTVP with different node densities we use different Scenes, shown at Table 3.3.

Figure 3.30 shows the average number of packets DC-ARQ Cooperated as a function of cars downloading. As expected, it shows that for the denser scenes there are higher cooperations, while for the sparser scenes the cooperations are low. And as a function of the number of downloading nodes the cooperations do not change.

Figure 3.31 is more interesting and difficult to explain, It is unexpected that at first the cooperations increase when the $\delta$ increases, it seems that if more nodes need cooperations then the cooperations received by a node should decrease because they will be shared by all. But what happens is that when there are few nodes
3.4. Delay Tolerant Vehicular Protocol

downloading, a node has the AP coverage to itself, and with all the bandwidth it downloads a bigger part, probably doing a complete round to the file. Then there will be no blocks completely unreceived. DTVP cooperation works with a granularity of blocks giving some of them to the cooperators, so cooperations will be less efficient because they carry less packets unreceived.

When the \( \delta \) increases, the bandwidth of an AP may be shared among more nodes, so they need to visit more APs to download the complete file, and there will be more opportunities to cooperate, and completely new blocks may be cooperated in this opportunities.

When the \( \delta \) increases more, it becomes more difficult for a node to DTVP-cooperate, because it is more probable that there are some nodes directly downloading from the AP, and then cooperations are avoided.

If the scenario is denser the behavior is more drastic. With a a high car density and low \( \delta \), there will be more cooperators than in a lower density scenario,
but when $\delta$ increases the system will collapse earlier. In a low density scenario, even if all but one of the nodes were downloading, when it crosses an AP may be alone and will have an opportunity to cooperate with several nodes. That is why in low density scenarios, even that the $\delta$ increases the system does not collapse.

![Figure 3.32: Average # of APs used to Download a 10MB file using DTVP](image1)

Figure 3.32 shows the average number of APs used to download as a function of cars downloading for all the scenes, it shows that scenes A, B and C have a lower increase while scene D suffers a high increase, because it reaches the capacity of the network and DTVP could do nothing to improve it, only DC-ARQ cooperations help. Then if the number of downloading nodes doubles the number of APs used will do the same.

![Figure 3.33: Average # of APs used to Download a 10MB file without coop, using DC-ARQ and using DTVP](image2)

Figure 3.33: Average # of APs used to Download a 10MB file without coop, using DC-ARQ and using DTVP
Figure 3.33 and 3.34 show for two different scenes, how the DTVP improves the performance in comparison to not using cooperation and using only DC-ARQ cooperation. It shows that when the $\delta$ increases the difference is higher and it is even higher for denser scenes, because the saved retransmissions improve the overall throughput.
3.4.3 Conclusions

We have seen that in this framework, even that it is not easy due to the *Coupon collector’s problem*, each node achieves to reduce the number of APs used for the download, or what it is the same, reduces the delay to get it. But in addition to the node improvement, it also reduces the number of retransmissions, thus improving the overall performance allowing the network to scale better in scenes with high density.

We have showed that carry-and-forward in opposite direction is a good mechanism to fit with DC-ARQ, because they complement each other. While DC-ARQ achieves its best performance in high node density scenes, carry-and-forward in opposite direction achieves its best performance in low node density scenes.

The only conflict that may appear between them is caused by coordination issues. It is possible that a node leaves an AP without acknowledge the last blocks, and in addition, the AP could not know which packets of the whole blocks transmitted will be cooperated. So a block may be completely sent and received but if it was not acknowledged it could be chosen for a DTVP cooperation. Network Coding could help to avoid or soften this issue. Using Network Coding, beacons could change the block bitmap, by a solution that would behave similar to the Complete Bitmap, without using an excessively long bitmap. Beacons only need to carry the number of packets received for each block. Then as the last beacon received offers a complete view of the download status, that would be more accurate than that the actual framework could offer. The AP with this information, and knowing which packets have been sent since the last beacon, and how many DC-ARQ cooperators it will have, could foresee with a certain probability how many packets of each block would be necessary for the DTVP cooperations.

We have also seen that even in scenes where nearly all the nodes download, it is possible to achieve many carry-and-forward in opposite direction cooperations, if car density is low. When a not downloading node crosses an AP possibly it will be alone, allowing it to use all the bandwidth to cooperate with many destinations.

In addition we must say that understand some of the plots of this section and tune some parameters, like the beacon period, could have been difficult without the help of the previous sections.
Chapter 4

Summary, Conclusion and Future Work

This master thesis has proposed a DTVP framework, showing how different low coordination end-to-end ARQs perform in a Delay Tolerant Vehicular Network. And after evaluating them alone and in combination with a DC-ARQ we have proposed one for the DTVP framework, to complement the carry-and-forward in opposite direction, and we have analyzed how they improve global and local performance parameters.

We combat the harmful effects of Delay Tolerant Vehicular Networks, applying end-to-end modifications at the client and at the server, and modifications in the network layer, to apply a custodian based solution.

We identified the need of a mechanism to dynamically control the beaconing load. In high density scenarios a high beaconing load results in a high amount of packet collisions, which decreases the throughput, while in low density scenarios a low beaconing load increase coordination issues, (e.g. retransmitting duplicated packets) and nodes spending also more time to notice AP coverage.

DC-ARQ and DTVP cooperations are done for free at the dark areas where there are no AP coverage, or in an AP coverage if there are no node downloading. Thus, avoiding to interfere with current downloads, because in this scenarios bandwidth is a very scarce resource.

We have seen that synergy is gained when using both protocols together since while DC-ARQ achieves its best performance in high node density scenes, carry-and-forward in opposite direction achieves its best performance in low node density scenes.

Our study included a detailed implementation and simulation analysis using ns-2.

In the experiments we showed a recovery on the order of 50% of the packets lost in urban areas using only 2 cooperators.

Furthermore, according to the results obtained from the evaluation, we can conclude that we accomplished our goals with reasonable low complexity solutions.
The experiments of DC-ARQ were published in [1], the first simulations of DC-ARQ were published in [2], and now we want to summarize all this previous work and the progress done in this Master Thesis in a new paper. Adding the carry-and-forward in opposite direction and merging everything in a Delay Tolerant Vehicular Protocol framework.

We have done some experiments about cooperation between car crossings in a urban scenario, but more experiments of car crossings at higher speeds must be done to recognize which patterns and limitations has this scenario and to validate the feasibility of the results of the simulation of cars crossing in opposite direction.

Moreover, although we have shown satisfactory results, and we have already done a lot of work, there are still several ideas to improve global and local performance parameters.

Network Coding will fit very well in this delay tolerant vehicular scenario. It would make easy several issues like coordination, problems related to the Coupon collector’s problem, simplify policies for choosing which packets of which blocks must be transmitted to which cooperator and how. But as a DTVP framework that integrates the end-to-end ARQ, a DC-ARQ, carry-and-forward in opposite direction and Network Coding would be very complex to analyse we started studying how the DTVP framework without Network Coding works. The next step is to integrate Network Coding in the DTVP framework.

We plan to write down the main results in a paper to be sent to a conference.
Bibliography


[34] “SUMO.” Online: http://sumo.sourceforge.net


[38] “OPNET.” Online: http://www.opnet.com/

[40] “Straw.” Online: http://www.aqualab.cs.northwestern.edu/projects.html

[41] “EPFL TraNS.” Online: http://wiki.epfl.ch/trans


[50] “Click Modular Router.” Online: http://read.cs.ucla.edu/click/