sensor network).
The results of the simulations (figures: 4.19, 4.20, 4.21 and 4.22) show clearly that the performance trend of both Ns2 and Qualnet is almost linear. The flatters on the both graphs is mainly related to the increasing number of nodes in a relatively small simulation area (as the nodes, randomly moving in the area, come too close each other, the high number of collisions can raise up the number of the events the simulator has to handle). Although the performance trend in Ns2 and Qualnet is almost linear, anyway it should be noticed the big difference in terms of absolute values: if considering the trend line, Ns2 needs almost 900 seconds to perform the same simulation that Qualnet handles in 63 seconds and this is primarly due to the different implementations of the core functions. It should also be highlighted that Qualnet execution times are shorter even it performs more accurate simulations at physical and mac layers.[1]

[1]SNT whitepapers prove Qualnet precision and detailed level of simulation in their whitepapers (they cannot be referenced in this site). For further information, please refer to the SNT website at: *http://www.scalable-networks.com/*.

Chapter 5

Data dissemination survey

This chapter will survey the different types of information exchange adopted in vehicular networks with common practices and methodologies that have been considered in research literature (e.g.: *opportunistic exchange of resources between vehicles*, *vehicle assisted data delivery*, *cooperating downloading of information, etc.*) with a special emphasis to the network coding technique.

5.1 Types of information involved in the dissemination process

The different types of information that need to be communicated and then shared in a vehicular network can be roughly classified in four categories:

- *Safety information*
  Safety information is probably the most important information type that are communicated in a vehicular network (109), as motor vehicle crashes cause more than 120.000 victims in Europe and 1.2 million worldwide[1]. As you may recall, in chapter 2 we have introduced DSRC (Dedicated Short-Range Communications), a variant of the 802.11a technology designed to support V2V and V2I and conceived by the Federal Communications Commission (FCC) to *"increase traveller safety, reduce fuel consumption and pollution and continue to advance*

[1]Fact Sheet EURO/03/04 (Copenhagen, Rome, 6th April 2004) of the World Health Organization, ONU.

the nation's economy". DSRC will support safety-critical communications such as collision warnings as well as other valuable Intelligent Transportation System applications (i.e.: electronic-toll-collection (ETC), realtime traffic advisories, digital map updates, etc.). As it concerns safety messages, FCC recognizes *safety messages* and *safety-of-life messages*, which have the highest priority, whether originated by vehicles or roadside transmitters. Thus, DSRC can be used to prevent collisions between vehicles by providing important informations to the drivers about whether the vehicle ahead is braking, if the speed is too high or the distance to other vehicles (or objects, more in general) is getting too close. It is valuable to remark that the US Department of Transportation and the Vehicle Safety Communications Consortium (VSCC) have identified eight applications that rely on security informations(81):

1. *Curve Speed Warning*
   The Curve speed warning systems use a combination of GPS and digital maps to assess threat levels for a driver approaching a curve too quickly: if the driver enters a curve at a speed that will not allow him to drive through the curve safely, he will automatically receive a warning. This warning can be an acoustic signal or a visual symbol that appears in the navigation system or directly on the instrument panel.

2. *Traffic Signal Violation Warning*
   The traffic signal violation warning system is designed to warn a vehicle when the system detects that the vehicle is in danger of running the traffic signal. The system will make a decision (sending a warning message to the vehicle) based on the traffic signal status and timing and the vehicles speed and position. Some other factors are considered in such situations, such as the road surface and weather conditions.

3. *Emergency Electronic Brake Lights*
   This application alerts the driver when a preceding vehicle performs a severe braking maneuver by trasnmiting the alert notification with DSRC. Surrounding nodes that receive the message must then discern if the event is relevant, since, as we have already said, they can ignore warnings from vehicles traveling behind, far ahead, or in the opposite direction.

4. *Pre-Crash Warning*

   This system is designed to reduce the severity of an accident. Depending on the system, they may warn the driver, precharge the brakes, retract the seat belts removing excess slack and automatically apply partial or full braking to minimize the crash severity.

5. *Left Turn Assistant*

   The left turn assistant system helps a driver to make a safe left turn at signalized intersections with no left turn arrow phase. The system will collect data about vehicles coming from the opposite direction that are approaching the intersection, using sensors located at the intersection or DSRC communications, and then inform vehicles that are going to make the left turn. One way to implement such system is by having data collected continuously, and when there is a vehicle with its left turn signal on, the system will send a message to that vehicle about the traffic traveling in the opposite direction of the vehicle. The other way is to have an in-vehicle system that sends a request to be notified about the traffic in the opposite direction when the left turn signal is activated. The system will then collect data about the traffic in the opposite direction and send a message to the vehicle that requested the information. In both ways, the driver will be informed and warned about the traffic coming in the opposite direction. The is system is different to most of the other intersection collision avoidance systems because it can use both V2I and I2V communications.

6. *Lane change Warning*

   These in-vehicle electronic systems monitor the position of a vehicle within a roadway lane and warn a driver if it is unsafe to change lanes or merge into a line of traffic. These systems are rearward-looking and assist drivers who are intentionally changing lanes by detecting vehicles in the driver's blind spot.

7. *Stop Sign Movements Assistance*

   The stop sign movement assistant system is designed to avoid accidents at stop sign intersections. The system uses data collected by sensors or DSRC communications to inform the driver if it is unsafe to go through an intersection, so it warns drivers if there is traffic coming through the intersection at the same time. This system uses both V2I and I2V communications.

8. *Cooperative Forward Collision Warning*
This warning system is designed to warn drivers that are approaching the same intersection if there is a possibility of an accident occurrence. Data about all vehicles approaching an intersection are gathered using sensors and/or DSRC communications and then processed to see if there is a chance of a collision at the intersection. If there is such a situation, messages will be sent to vehicles that might be involved in this unsafe situation. The data required by the system includes information regarding the vehicles that are approaching the intersection and other data about the road. The data needed about vehicles are position, velocity, acceleration, and turning status. The data needed about the road are shape and surface. Data about the weather status is helpful in calculating the chance of a collision occurrence.

The communication requirements of these eight applications are shown in table 5.1. Note that communication frequency ranges from 1-50 Hz, the size of the packet ranges 200-500 bytes, and the maximum communication range spans from 50-300 meters. Further, high-level Data Element requirements are specified and several of these Data Elements (e.g. position and heading) are needed by multiple applications. Responding to these identified applications, the "Dedicated short range message set" defines over seventy vehicle Data Elements (e.g. heading, acceleration (with varying precision: 4bit, 8bit, 16 bit), headlight status and brake status). Of these Data Elements, thirty of the most commonly used elements are selected for a common message set, as listed in table 5.2. However, very few of these applications have actually been implemented or fully developed. Hence, the exact usage characteristics of the safety messages were in flux at the time the safety messages were being defined and standardized.

Clearly, in each of the said applications, the time taken by the driver of a motor vehicle to react to warnings has to be considered while delivering information(109), thus introducing the human component into the picture as well as the traffic density and the environment, factors which influence the delivery of safety messages.

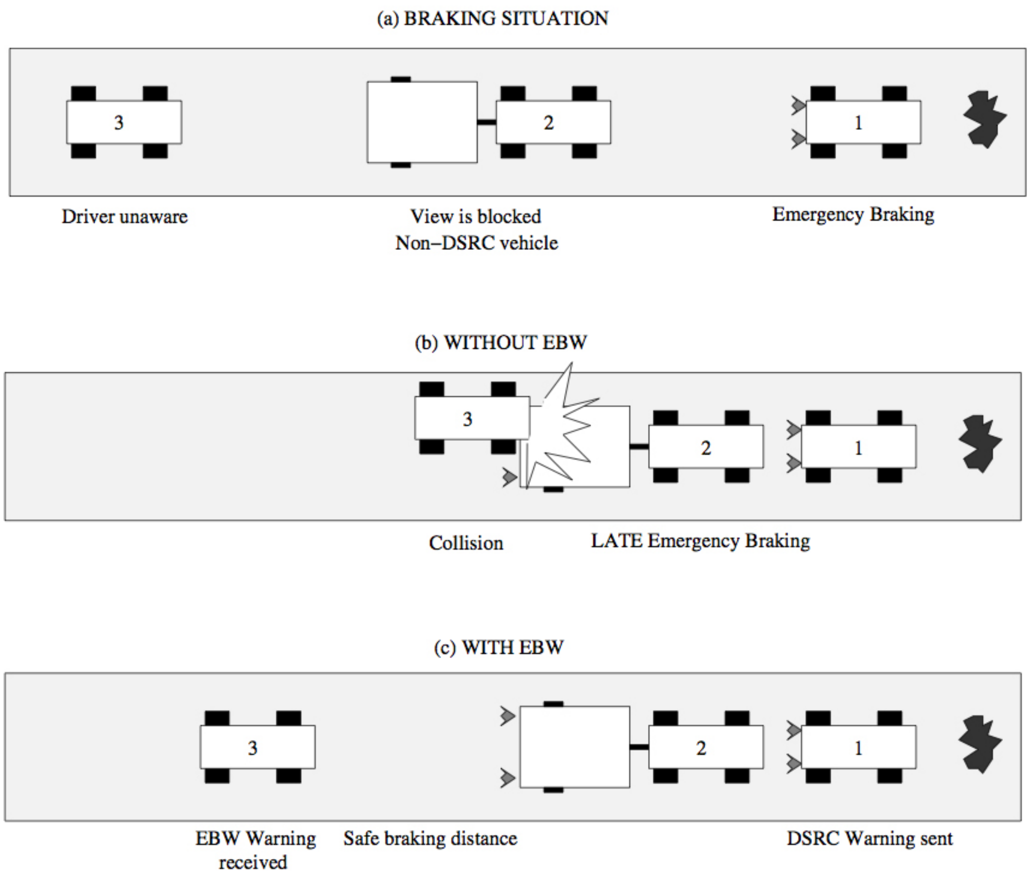

Figure 5.1: EBW - Emergency Brake Warning. The EBW alerts the driver behind by augmenting the brake-lights notification system when a preceeding vehicle performs a severe braking maneuver.

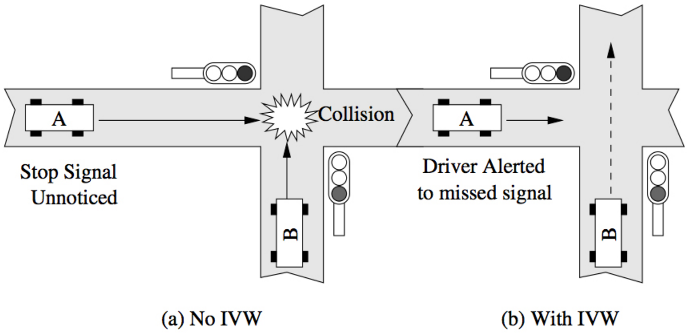

Figure 5.2: IVW - Intersection Violation Warning. The IVW warns the driver if violating a red light seems imminent. Other vehicles approaching the traffic lights are also warned that a close vehicle issued a warning message.

- *Traffic information*

  Congestion of roads is a very important information to share on vehicular networks in order to let personal electronic navigators to select the best route and avoid traffic. This can be extremely useful in case of car crash or even under emergency circumstancies because mobile nodes can be informed in advance on the congestioned area so that service vehicles can quickly reach the destination. Of course, this kind of messages are without any doubt useful for safety and efficiency but less time-critical than the "life safety" ones discussed earlier. With respect to infotainment and content information, this class of data still has higher priority.

- *Infotainment information*

  With infotainment information we intend all that kind of messages about entertainment and, more in general, useful messages to the driver. An example can be:

  - locating the closest coffee shop
  - locating the closest cinema or mall
  - locating the fuel station which offers the best price in that area
  - locating the closest avaliable parking spot
  - many and many others...

Application	Comm. type	Freq.	Latency	Data transmitted	Range
traffic signal violation	I2V one-way, P2M	10 Hz	100 ms	signal status, timing, surface heading, light pos., weather	250 m
curve speed warning	I2V one-way, P2M	1 Hz	1000 ms	curve location, curvature, bank speed limit, surface	200 m
emergency brake lights	V2V two-way, P2M	10 Hz	100 ms	position, deceleration heading, velocity	200 m
pre-crash sensing	V2V two-way, P2P	50 Hz	20 ms	vehicle type, yaw rate, position heading, acceleration	50 m
collision warning	V2V one-way, P2M	10 Hz	100 ms	vehicle type, position, yaw rate heading, velocity, acceleration	150 m
left turn assist	I2V and V2I one-way, P2M	10 Hz	100 ms	signal status, timing, position direction, road geom., heading	300 m
lane change warning	V2V one-way, P2M	10 Hz	100 ms	position, heading, velocity acceleration, turn sign. status	150 m
stop sign assist	I2V and V2I one-way	10 Hz	100 ms	position, velocity heading, warning	300 m

Table 5.1: Eight high priority vehicular safety applications as chosen by NHTSA (71) and VSCC (*Vehicle Safety Communications Consortium*), which is a consortium born in May, 2002 with the mission to facilitate the advancement of vehicle safety through communication technologies (the members are: BMW, DaimlerChrysler, Ford, GM, Nissan, Toyota, and VWGM, VW). Communication frequencies ranges from 1-50 Hz and the maximum range span from 50-300 meters. Note that P2M represents "point-to-multipoint", I2V "infrastructure-to-vehicle" and V2I "vehicle-to-infrastructure".

- *Content*

  This kind of information differs from the others especially in the quantity of data exchanged between nodes and recent research works in vehicular networks have focused on content distribution as multimedia data are becoming essential in everyone's life(55)(68)(28); this means the main assumption is that vehicles are interested in obtaining extremely large files (e.g. videos and movie clips) that cannot be shared through a limited transactions with either the infrastructure or the nearby mobile nodes. The discovery and distribution of large quantities of information is still a challanging problem especially in a dynamic environment such as a vehicular network.

5.2 Data dissemination techniques

The process of information dissemination determines a bunch of problems because of many reasons. At first glance, we have to consider the network size but also the vehicles' speed and the patchy and intermittent connectivity between mobile nodes; moreover, there is another problem which can severely affect the entire process: *latency requirements*(109). As a consequence, content information has to be quickly discovered and shared among nodes.
In the research literature, many methods of information delivery are presented and, generally speaking, we can distinguish the following data dissemination approaches:

- *opportunistic:* information is pulled from other vehicles or the infrastructure as a target vehicle encounters them(108);
- *vehicle-assisted:* a vehicle carries information with it and delivers it either to the infrastructure or to other vehicles when it encounters them. This process involves mobility in addition to wireless transmissions in order to disseminate the information(113);
- *cooperative:* vehicles can download partial units of some content and then share them afterwards to obtain the complete content. This method is particularly suitable for content dissemination (where the amount of information is rather important in terms of file size) and it was adopted to develop our dissemination protocol based on rateless codes(68) (discussed in detail in chapter 6).

In addition to these dissemination methods, it is important to take into consideration the *spatial and temporal constraints*(108)(105), that is evaluating how far the information should be disseminated(109). For example, in case of traffic and service information, aging with time and distance is suggested in (108), where data, opportunistically pulled from neighboring nodes, are "marked" with a time stamp and a location stamp. As the mobile node moves father away from where the information is relevant (or as time elapses), the information itself is aged or eventually purged, thus allowing vehicles to mantain up-to-date information without penalizing memory

or other resources. Because of the spatio-temporal relevance of information, a piece of information tends not to propagate beyond a specific boundary; in time, a given piece of information would propagate very quickly till it reaches a maximum number of copies and then it would also rapidly decline from that point.
As it concerns the content distribution in vanets, several solutions have been discussed in the last few years. The basic idea is that vehicles download fragments of files when they have access to the roadside and then share them with other mobile nodes on the way. Although it seems a simple approach, there are many issues related to the sporadic nature of the connectivity.
Sometimes, the content is partially downloaded from a roadside base station, which also provides a list of other vehicles that have parts of that file, like the solution proposed by Nandan et al. in (68). On the contrary, in the work we presented at WCNC'09 (20), the resource is entirely downloaded by the infrastructure (or other vehicles) and then shared among mobile nodes without any provided assistance or list of any kind by the base station. In the first scenario, each vehicle "gossips" about what part of the content it has, while in the second, it answer to a specific request of other nodes.
In *BitTorrent* (68), a closest-rarest strategy is adopted where the rarest piece of the fiel is downloaded first, but it is decentralized and employs proximity information in order to improve the overall performance. This is because, in general, multihop communications degrade the performance of the information delivery rapidly as the number of hops increases. To overcome the said problem, we can cite the work done in *CodeTorrent* (cfr. 5.4.2) (55), where *network coding technique* and *mobility assistance* along with gossiping (as in (68)) can be used with one single hop distribution of content, thus further improving performance.

5.3 Information sharing via Network Coding technique

5.3.1 Introduction to Network Coding

In scenarios where delay-tolerant applications are implemented, different approaches can be applied in order to reduce as much as possible the number of transmissions required to deliver some amount of data, thus improving the overall performance. This goals especially interesting in VANETs[1], where the reduction of the traffic plays an

[1]More in general, it is also relevant in pervasive systems.

important role in guaranteeing better performance. To achieve this aim we can adopt recent network paradigms which introduce in the network protocols some kind of intelligence to further increase the efficiency in data transmissions. Data dissemination is one of the most important services in such networks: it can be used to disseminate control data such as routing information, device status, performance metrics and, more in general, any kind of interesting content. Furthermore, data broadcasting could also be useful when a lot of users want to share some kind of data in a distributed way: downloading the same files, sharing knowledge about the network services and so on (as we saw in *content data dissemination*). In all these scenarios, similar information, usually distributed over the whole network, has to be exchanged by the users.
The heterogeneity of the devices together with the wireless characteristics of the networks, make the definition of an efficient data dissemination scheme a challenging issue: the main goal here is to guarantee a limited usage of the network resources in order to increase the throughput. This translates in defining a data dissemination scheme which is able to deliver a lot of data at the minimum cost in terms of number of transmissions. We know that the transmission phase is particularly expensive for wireless devices and, in addition, high traffic leads to unsatisfactory performance due to the contention mechanisms used in wireless communications. Thus, reducing the number of transmissions to deliver data to multiple users is a very important goal in such networks. Furthermore, if we think of VANETs, the reduction of transmission is essential because of the mobility: we could say that the communication has to be *"effective"*.

A lot of studies, in literature, follow this direction by proposing different solutions. Recently, a technique known as *in-network data aggregation* has been developed for wireless sensor networks (26). It consists of some data processing along the network in order to combine packets and reduce the amount of information to be spread. However, since the data processing performed over the packets is generally lossy, in-network data aggregation is useful when nodes are interested on some average measures of the required information. For instance, it is particularly useful in wireless sensor networks for environmental monitoring to gather some information about the average temperature in a specific area and so on. This kind of approach, however, can not be applied for lossless data dissemination services, such as those generally related to the VANETs. However, the innetwork data aggregation approach suggests that for an efficient data dissemination we have to go to a specific direction: introducing some intelligence at

the network layer or even at application layer (as we did in our protocol (20)) in order to carry more information by transmitting the same number of packets, in some how coded .
The Network Coding technique is one of the promising approach because it allows to achieve the said goal without loss of information. In the following paragraphs we study how to apply network coding paradigm in data dissemination in order to make the data delivery more efficient in terms of energy and throughput.

5.3.2 What is Network Coding

Network coding is a new research area that can offer interesting applications in practical networking systems. With network coding, intermediate nodes may send out packets that are linear combinations of previously received information. There are two main benefits of this approach: potential throughput improvements and a high degree of robustness. Robustness translates into loss resilience and facilitates the design of simple distributed algorithms that perform well, even if decisions are based only on partial information.

The concept of network coding is firstly introduced in (9) by Ahlswede et al. and it can be formulated in various ways at different levels of generality. A simple definition is reported next:

"Network coding is a particular in-network data processing technique that exploits the characteristics of the wireless medium (in particular, the broadcast communication channel) in order to increase the throughput of the network."

In other words, it is a packet dissemination strategy which can be used to improve the throughput, thus guaranteeing high performance. In contrast to the store and forward paradigm, network coding implements a more complex *store, code, and forward* approach where each node stores the incoming packets in its own buffer, and, at transmission time, sends a combination of the stored data. To successfully decode, say, K packets, a node must collect K independent combinations. It can provide the highest gains in multicast or broadcast networks. More specifically, network coding can typically achieve higher transmission rates than separate unicast transmissions when information sources transmit to multiple destinations or to all nodes in the network. The basic mechanism of network coding is introduced in (9) through a simple example which we report for readers ease: consider the acyclic network depicted in Fig.5.3, with 6 nodes and a single source S . Time is subdivided in slots in a TDMA fashion and wireless links can carry a single flow at a time, i.e., either flow $b1$ or $b2$ can be transmitted during a time slot over a specic link. The problem to be solved is to send two bit flows $b1$ and $b2$ to both destinations Y and Z by exploiting the network at its maximum capacity. In other words, the flows are to be multicast to both destinations in the most efficient way.

As a first attempt to solve the problem, one might devise a simple transmission schedule which consists of alternating two distinct transmission modes between even and odd slots as follows. In even slots we may let links $T-X, X-Y, T-Y, X-Z$ carry flow $b1$, and links $S2-U$ and $U-Z$ carry flow $b2$. Hence, during even slots node Y receives flow $b1$ only, whereas node Z receives both flows.

This transmission schedule is depicted in Fig.5.3.A. During odd slots, instead, the situation may be reverted according to the transmission schedule in Fig.5.3.B. In such a case, node Y receives both flows and node Z receives flow $b2$ only. Overall, this strategy leads to an average throughput of 1.5 flows per slot. A different way to attack the problem is provided by network coding and is shown in Fig.5.3.B. This time, node X derives from flows $b1$ and $b2$ the exclusive-OR $b1 \oplus b2$, which is finally passed to both destinations Y and Z . Destination nodes can now decode both flows by re-applying the XOR operation. This strategy obtains a throughput of 2 flows per slot, which is the maximum achievable in the above problem. We observe that the exclusive-OR is a simple form of coding. If the same objective is to be achieved by store and forward techniques (simple replication of the incoming data flow), then at least two subsequent slots are needed to deliver $b1$ and $b2$ to both destinations. With coding, the two flows can instead be delivered to both Y and Z in a single slot. It is therefore apparent that coding, besides offering advantages in terms of throughput, may also decrease the latency. Note, however, that a drawback of the above example is that the coding/decoding scheme has to been agreed upon beforehand. That is, node X must know in advance that it has to process the received flows by means of some coding strategy. While this is acceptable to present the advantages of coding by means of the above example, it has profound implications on the actual applicability of such

a technique in distributed and therefore uncoordinated networks (these implications together with possible solutions will be explored in the next paragraph).
The Network Coding technique offers another important benefit: *the increase of the information capacity, which is the capacity of the link (between two nodes) to transport innovative information.*
Referring to our above example, channel $X - Y$ and channel $X - Z$ have a transport capacity of a single flow per time slot. However, if we transmit a coded information over such a link, then *we obtain the higher transmission rate of 2 data flows per time slot.* In other words, the information capacity with coding is higher[1].

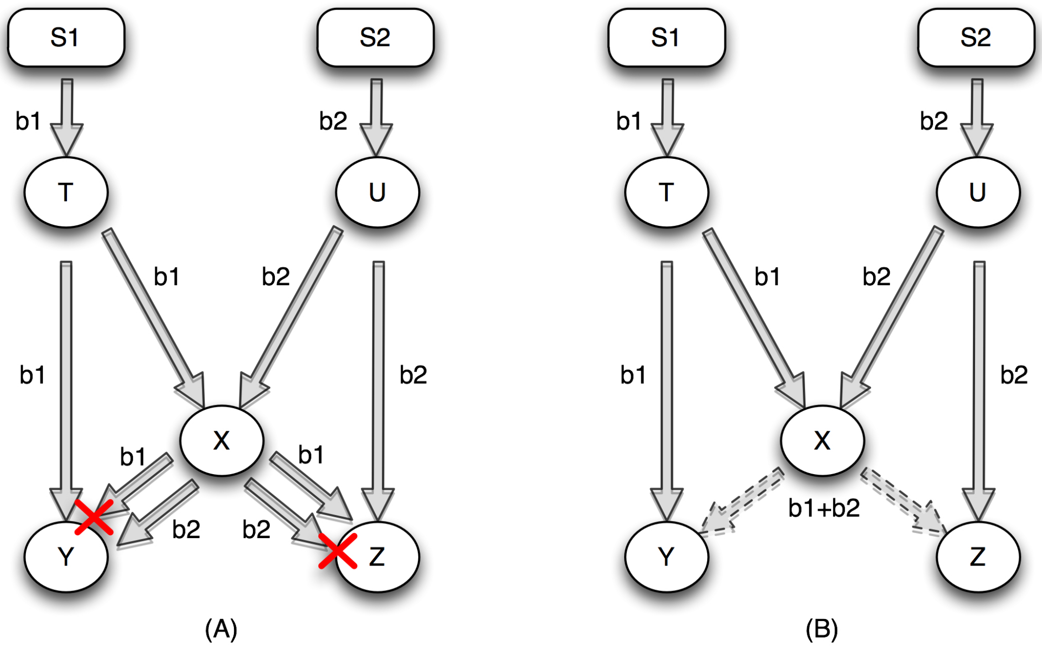

Figure 5.3: Network Coding. Simple example of Network Coding mechanism.

To better explain this idea, please consider the additional example reported in Fig. 5.4, which is closer to the wireless ad hoc communications. It represents a two hops network where each of *nodeA* (NA) and *nodeB* (NB) have to exchange a data packet, $b1$ and $b2$, respectively. Both the routing paths, namely the one from NA to NB and the other way around, go through *NodeC* (NC) which has to forward two packets.

[1]This concept is extremely important, as it will be considered again in our protocol, VDRP.

The first transmission phase, i.e., both NA and NB send their own packets to NC, which is equivalent to the classical communication paradigm and to network coding (see Fig. 5.4-b). Then, according to the classical communication paradigm, we need other two transmissions to deliver both $b1$ and $b2$ to the corresponding destination (see Fig. 5.4-c). With network coding, instead, we can use only one additional transmission by allowing NC to transmit the XORed version of packet $b1$ and $b2$ to both NA and NB as reported in Fig. 5.4-d. At the reception of this packet, NA (or NB) can decode packet $b2$ (or $b1$) by subtracting its own packet from the received one. This example shows how network coding is particularly effective whenever there are overlapping data flows, because it can exploit both the broadcast nature of the channel and the coding process to simultaneously deliver different packets to multiple users. After having briefly discussed the possible advantages offered by network coding, we now need to make things practical. In fact, it is actually infeasible to have pre-defined coding rules for every node in the network, as this would require full network knowledge. More than this, in distributed ad hoc scenarios, we need to cope with many constraints. For instance, nodes are not synchronized and, as a consequence, coding operation should not depend on time synchronization. Also, as messages travels through the network, they are exposed to delays (mainly due to processing at intermediate nodes and channel contention mechanisms). Finally, too complex coding schemes are actually to be avoided as they might be infeasible for resource limited wireless devices. Possible solutions to these issues are discussed in the following paragraphs.

5.3.3 Linear Network Coding

Network codes that involve only linear mappings are of particular interest as they can be executed at low computational cost. The word *"linear"* clearly let us imagine that the output flow of a given node is obtained as linear combination of its input flows (29). The coefficient of the combination are, by definition, selcted from finite Galois field[1] and the information traversing a non source node has the following property:
"The content of any information flowing out of a set of non source nodes can be derived from the accumulated information that has flown into the set of nodes"

[1] In abstract algebra, a finite field or Galois field (so named in honor of Evariste Galois) is a field that contains only finitely many elements.

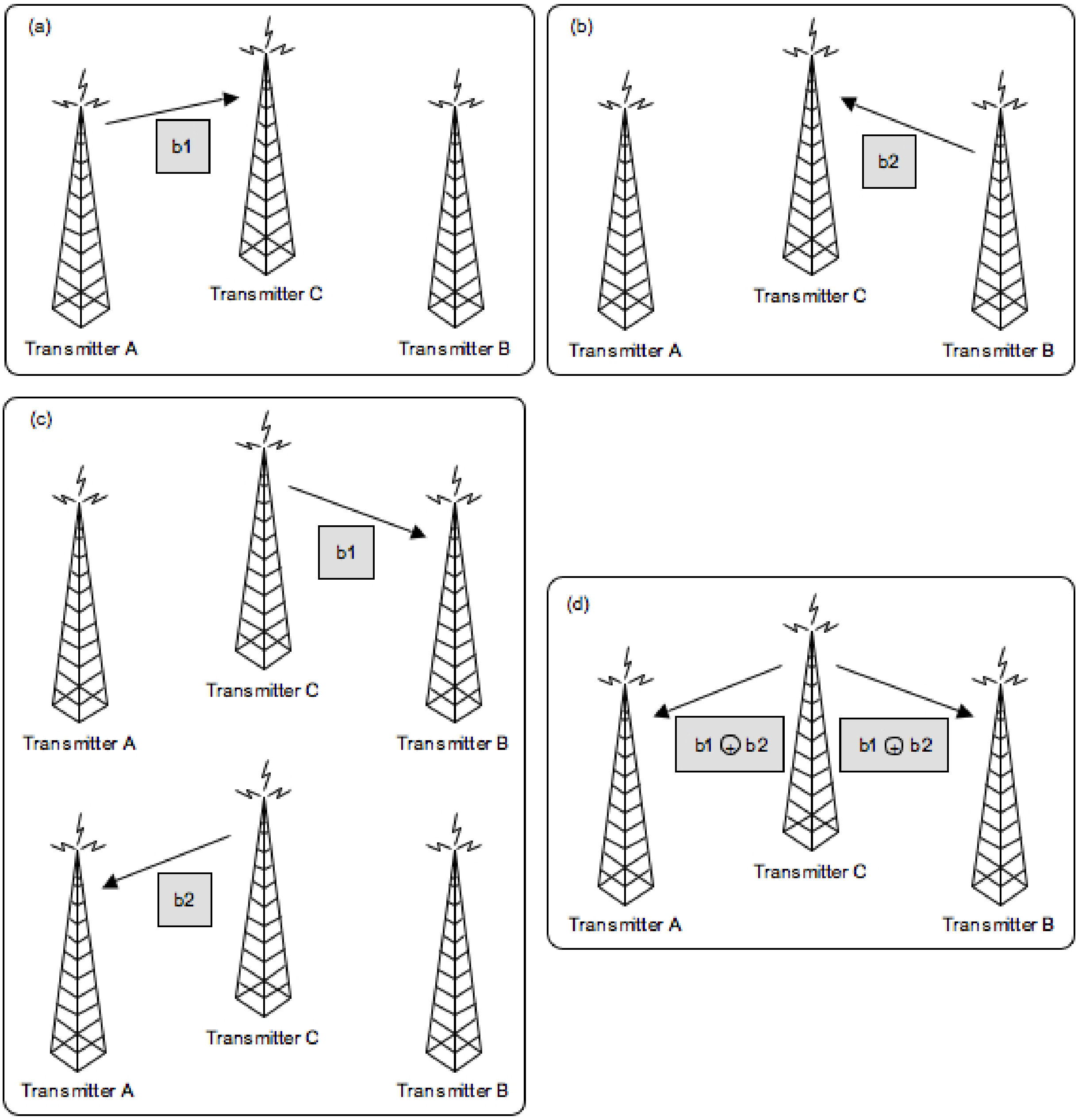

Figure 5.4: Network Coding. The figure shows the transmission of two data packets with the classical communication paradigm compared to the network coding technique. If two nodes are communicating with an intermediary node, they can benefit of this coding technique, thus optimizing the use of the channel and gaining higher throughput (in the example, nodes A and B want to communicate using node C as relay agent; while with the classical paradigm node C has to use the channel twice in order to send the flows $b1$ to B and $b2$ to A, with the network coding technique only transmission of flow $b1 + b2$.

Assume that each packet consists of L bits. When the packets to be combined do not have the same size, the shorter ones are padded with trailing $0s$. We can interpret s consecutive bits of a packet as a symbol over the field $\mathbb{F}_{2s}$, with each packet consisting of a vector of L/s symbols. With *linear* network coding, outgoing packets are linear combinations of the original packets, where addition and multiplication are performed over the field $\mathbb{F}_{2s}$. Clearly, one of the reasons for choosing a linear framework is that the algorithms for coding and decoding are simple and well understood.
It is important to consider that linear combination is not a concatenation: if we linearly combine packets of lenght L, the resulting encoded packet also has size L. An encoded packet generally carries information about several original packets, but in contrast to concatenation, just by itself it does not allow to recover any part of the original packets. Once can think of linear *network coding as a form of information spreading.*

5.3.4 The coding and decoding process

The coding process can be easily explained as follows. Assume that a number of original packets $M^1, ..., M^n$ are generated by one or several sources. In linear network coding, each packet in the network is associated with a sequence of coefficients $g_1, ..., g_n$ in $\mathbb{F}_{2s}$ and is equal to:

$$X = \sum_{i=1}^{n} g_i M^i \tag{5.1}$$

The summation has to occur for every symbol position, i.e., $X_k = \sum_{i=1}^{n} g_i M_k^i$, where M_k^i and X_k is the k^{th} symbol of M^i and X respectively.
In the example of figure 5.4, the field is $\mathbb{F}_{2s} = \{0, 1\}$, a symbol is a bit, and the linear combination sent by S after receiving $M^1 = b_1$ and $M^2 = b_2$ is $M^1 + M^2$ (where the sign "+" is to be intended as the bitwise *xor*, the addition in $\mathbb{F}_{2s}$).
For sake of simplicity, we can assume that a packet contains both the coefficients $g = (g_1, ..., g_n)$, called *encoding vector*, and the encoded data $X = \sum_{i=1}^{n} g_i M^i$, called *information vector* (111). The encoding vector is used by recipients to decode the data, as explained later. For example, the encoding vector $e_i = (0, ..., 0, 1, 0, ..., 0)$, where the 1 is at the ith position, means that the information vector is equal to M^i.
Encoding can be performed recursively, namely, with already encoded packets. Consider a node that has received and stored a set $(g^1, X^1), ..., (g^m, X^m)$ of encoded packets, where g^j (and respectively X^j) is the encoding (respectively, *the information*) vector

of the jth packet. This node may generate a new encoded packet (g', X') by picking a set of coefficients $h_1, ..., h_m$ and computing the linear combination $X' = \sum_{j=1}^{m} h_j X^j$. The corresponding encoding vector g' is not simply equal to h, since the coefficients are with respect to the original packets $M^1, ..., M^n$; in contrast, straightforward algebra shows that it is given by $g'_i = \sum_{j=1}^{m} h_j g_i^j$. This operation may be repeated at several nodes in the network.

As it concerns the decoding phase, let us assume a node has received the set

$$(g^1, ..., X^1), ..., (g^m, ..., X^m) \tag{5.2}$$

In order to retrieve the original packets, it needs to solve the following linear system with m equations and n unknowns (M^i):

$$\{X^j = \sum_{i=1}^{n} g_i^j M^i\} \tag{5.3}$$

In order to recover all data, we need $m \geq n$, which means that the number of received packets needs to be at least as large as the number of the original packets. Conversely, the condition $m \geq n$ is not sufficient, as some of the combinations might be linearly dependent.

At this point, it should also be clear that one of the main problem in network coding is how to select linear combinations that each node of the network performs. A simple algorithm is to have each node in the network which selects uniformly and randomly the coefficients over the said field $\mathbb{F}_{2s}$, in a completely independent and decentralized fashion (36).

With random network coding there is a certain probability of selecting linearly dependent combinations which is mainly related to the field size 2^s (36). In (107), simulation results indicate that even for small field size (i.e., $s = 8$) the probability becomes negligible.

Alternatively, it is possible to use deterministic algorithms to design network codes: the polynomial-time algorithm for multicasting in (85) sequentially examines each node of the network and decides what linear combinations each node performs. Since each node uses fixed linear coefficients, packets only need to carry the information vector.

5.3.5 Benefits of Network Coding

One of the main advantages that sparked the interest in the network coding technique is that it can increase the capacity of a network for multicast flows. In (9) we can find an important theorem which states:

theorem. *Assume that the source rates are such that, without network coding, the network can support each receiver in isolation (i.e. each receiver can decode all sources when it is the only receiver in the network). With an appropriate choice of linear coding coefficients, the network can support all receivers simultaneously.*

Let us consider a network that can be represented as a directed graph, where the vertices correspond to channels and nodes to terminals, respectively, with M sources and N receivers (all interested in receiving data from sources): the said theorem states that when the N receivers share the network resources, each of them can receive the maximum rate, even if it were using all the network resources by itself.
Network coding may offer throughput benefits not only for multicast flows, but also for other traffic patterns such as unicast. Anyway, the throughput gains of network coding for the former can be very significant (85).
An interesting point is that network coding allows to achieve the optimal throughput when multicasting using polynomial time algorithms. In contrast, achieving the optimal throughput with routing is NP-complete: this is *the problem of packing Steiner trees in CS theory.* Thus, even when the expected throughput benets of network coding are not large, we expect to be able to achieve them using simpler algorithms.
Another very important benefit of network coding technique is given in terms of *robustness* and *adaptability.*

Intuitively, we can think that network coding, similarly to traditional coding, takes information packets and produces encoded packets, where each encoded packet is equally important. Provided we receive a sufficient number of encoded packets, no matter which, we are able to decode. The new twist that network coding brings, is that the linear combining is performed opportunistically over the network, not only at the source node, and thus it is well suited for the (typical) cases where nodes only have incomplete information about the global network state. Consider again Figure 5.4 and assume that A and B may go into sleep mode (or may move out of range) at random and without notifying the base station C . If the base station C broadcasts $b1$ (or $b2$), the transmission might be completely wasted, since the intended destination might not be able to receive. However, if the base station broadcasts $b1$ xor $b2$, or more generally, random linear combinations of the information packets, the transmission will bring new information to all active nodes.

Generally speaking, the network coding technique applied to wireless networks can improve throughput when two nodes communicate via a common base-station (cfr. 5.4). This setting can be extended to the case of multi-hop routing in wireless network, where the traffic between the two end nodes is bidirectional and both nodes have similar number of packets to exchange (106). In (106), the authors discuss a distributed implementation that works when transmissions are not synchronized and the wireless channel is lossy and has random delay. Overhearing a packet of a neighbor that is coded over information previously forwarded to the neighbor serves as a passive acknowledgment. This allows to make better use of transmission opportunities at routers that only have new packets buffered for a single direction. In this case, one of these new packets is combined with an old packet for the opposite direction, for which no passive acknowledgment has been received.
Another important gain in terms of network performance in wireless network given by network coding is the efficient implementation of many-to-many broadcast paradigm (used, for example, in route discovery), which may lead to significant energy savings(31).

In (32), authors simulate data dissemination and collection for sensor networks using a forwarding algorithm that sends out a packet with a certain probability, whenever a packet is received for the first time (flooding) or an innovative packet is received (network coding). When no data is disseminated and each node holds only its own packets, network coding and flooding obviously have the same performance. In case data is disseminated such that all nodes can decode all packets and a data collection phase is unnecessary, network coding offers a constant benefit in terms of required number of transmissions (31). Between these two extreme cases, network coding provides higher performance gains in terms of number of polled nodes.

For the scenario used in (32), authors vary the number of nodes in the network, as well as the network size, to have on average 12 neighbors per node. The total number of transmissions per node is limited and they count, how many nodes have to be polled by the data collector (sink) after the data dissemination phase, to obtain all data. Even

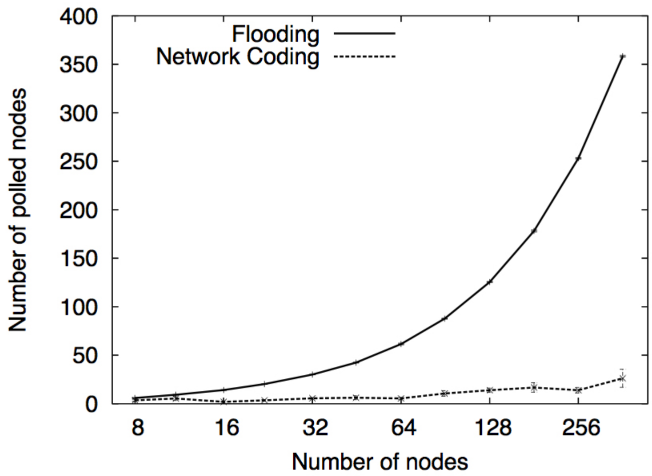

Figure 5.5: Network Coding benefits in data dissemination. The figure shows the number of nodes that have to be polled by the data collector to obtain all packets.

if nodes are allowed to transmit only once, network coding benefits a lot in very small scenarios but has little impact in larger networks, as shown in Fig.5.6.

As long as the allowed number of transmissions is signicantly below the number of nodes in the network, flooding needs to poll almost all nodes. Here, flooding with 256 transmissions per node achieves roughly the same performance as network coding with 4 transmissions per node. For more than 16 transmissions per node, network coding delivers all packets to the sink, such that no polling is necessary. For the next set of simulations, they set the number of packets each node is allowed to transmit to $0.25\sqrt{n}$ (n is the number of nodes). In that setting, flooding takes almost no advantage of the proactive data dissemination (graph of Fig.5.5). The number of polled nodes is reduced by slightly more than the number of packets that each node transmits. In contrast, with network coding the sink node needs to poll only a few other nodes to be able to decode all original packets.

Other scenarios that benefit from the said technique (but are beyond the scope of this document) are: *data gathering in sensor networks* (23), *network tomography* (30) and *network security* (37)(27)(34).

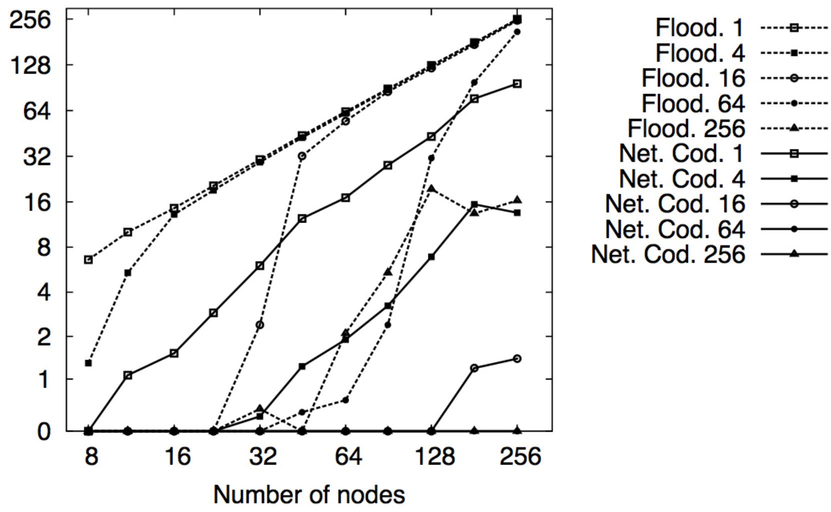

Figure 5.6: Network Coding benefits in data dissemination. Comparison between flooding and network coding: number of polled nodes for different numbers of transmission per node.

5.4 Example of Content distribution applications for VANETs

5.4.1 SPAWN (CarTorrent)

SPAWN (68) is a cooperative strategy for content delivery and sharing in VANETs, designed to improve client perceived performance of vehicular network as a whole. The key contributions of this protocol are:

- *a gossip mechanism* to propagate content avaliability information;
- *a proximity driven content selection strategy*, which takes into account the fact that TCP throughput degrades over multi-hop wireless connection;
- *leveraging the broadcast nature of wireless networks* to reduce redundant message transmission.

The basic structure of SPAWN is the same for every swarming protocol, that means peers downloading a file form a mesh and exchange pieces of the file amongst themselves. In order to perform an operation in SPAWN, there are three main phases to be taken into account:

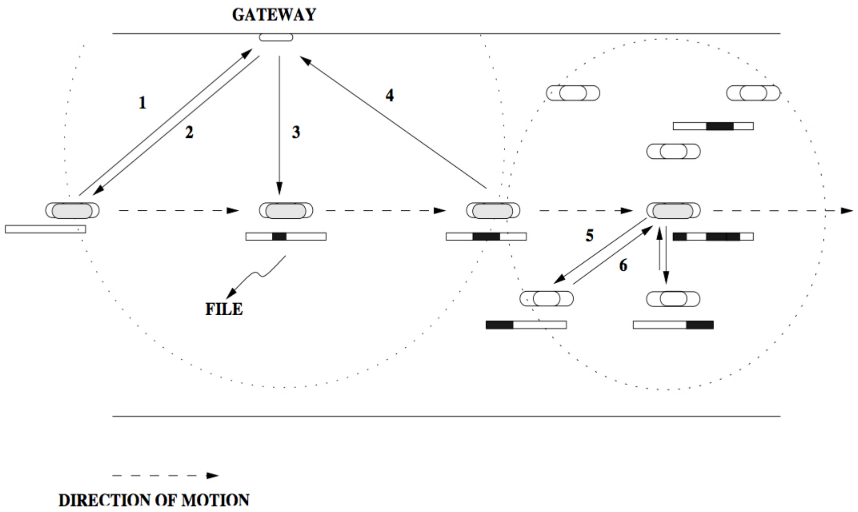

Figure 5.7: Evolution of a file in a node using SPAWN protocol. (1) A car arrives in the range of a gateway, (2) starts a download, (3) download a piece of a file, (4) after getting out of range, (5) stars to gossip with its neighbors about content avaliability and (6) exchanges pieces of the file, thereby getting a larger portion of the file as opposed to waiting for the next gateway to resume the download.

1. *Peer Discovery*

   A new node enters the vehicular network and requests the gateway for the particular file. If the gateway has the file in its cache, it starts uploading a piece to the node. Decision policies with respect to piece choice are:

   - *interested:*

     nodes are interested in obtaining that particular file;

   - *uninterested:*

     the nodes are downloading another file, thus they are not interested in this one;

   - *relays:*

     these nodes do not understand the SPAWN protocol and, then, they drop the received packets. However, as they run some routing protocol (i.e. to support the general vehicular network or other safety applications running on top), they will relay IP packets which are unicast through them, even those associated with SPAWN file transfer.

At this point, the node starts downloading pieces from the gateway while it is in range. The mechanism for peer discovery is decentralized by using the broadcast medium of the wireless channel to gossip information about the content availability at neighbors. In a mobile environment, gossiping provides a way to incorporate location awareness into the peer discovery scheme. Since TCP over multiple-hops suffers quite dramatically in the ad-hoc wireless scenario (33), a node is better off using unicast TCP connections with peers who are just 4 or 5 hops away. Hence in SPAWN, the centralized approach and the gossiping mechanism can be used in conjunction to construct the mesh of peers and update connectivity information. The gossiping mechanism is introduced to increase the robustness of peer list maintenance and discovery, in the presence of high degree of node churn, as well as the intermittent presence of gateways.

2. *Peer and Content Selection*
   SPAWN protocol does some intelligent Peer Selection based on the distance of the peer possessing a certain piece it intends to download. This information is gathered from the gossip messages but also from GPS enabled traffic safety messages that are likely to become standard applications running on vehicles in the future.
   As it concerns the content selection, a proximity-driven piece selection strategy, which uses several distinct strategies to choose the pieces to download, is used.

3. *Content Discovery and Selection*
   When two peers become neighbors, they exchange bitfields that describe which pieces they possess. When a peer fully receives a piece, it immediately notifies all of its neighbors. Thus, every peer always knows exactly which pieces its neighbors possesses. SPAWN uses UDP for delivery of gossip messages and TCP for content delivery.

SPAWN proves that the adoption of gossiping protocol in a VANET scenario offers several advantages. However, although analytical models have been developed, the simulation results are rather unrealistic because:

- the mobility model adopted in the simulation is too simple: cars mantain the same direction, almost the same speed and do not consider the cars around them;

- simulation parameters are themselves unrealistic: 5MB file size, 64KB data fragments and simplified version of the 802.11 DCF protocol.

5.4.2 CodeTorrent

CodeTorrent is a solution born to investigate the problem of data dissemination in vehicular networks. In more detail, this protocol tries to apply the P2P paradigm to VANETs by the use of network coding technique and mobility assisted data propagation as the main ingredient.
Clearly, if a node has to send a file F (seed node), creates and then broadcast to its 1-hop neighbors the description of the file. Further to this, F is splitted into n pieces and then encoded, according to the linear network coding technique. At destination, in oder to recover n file pieces $p1, p2, ..., pn$, a node must collect more than n coded frames carrying encoding vectors that are linearly independent of each other. Not only the seed node of a file, also every node which possesses any coded frame of the file and willing to share them periodically broadcasts (at a very low rate) to its 1-hop neighbors the description of the file. If a node has multiple files to share, multiple descriptions are packed into the least number of packets that can carry all of them and then transmitted. A request of coded frames may be accompanied by the null space vector which is a vector in the nullspace spanned by all encoding vectors of the frames stored in the local memory of the requesting node. On reception of such a request, a node transmits a coded frame only if there is in its local memory a frame with the encoding vector that is not orthogonal to the nullspace vector received with the request. Every node promiscuously listens to packets, i.e., a node receives a specific packet even the node is not the designated receiver, so that it can use them if possible. A node always overhears the packets carrying coded frames and treats the overhead ones as the coded frames transmitted specifically to the node. If an overheard coded frame is linearly independent of the coded frames in local memory, then a node stores it.

In CodeTorrent the average download delay decreases as mobility increases and, since this protocol is based on single hop data pulling and overhearing, mobility plays an important role such that *data dissemination latency could be reduced with increased mobility.* The main issues related to CodeTorrent (like almost every data dissemination protocol for VANETs) are the number of vehicles involved in the communications and