PROMETHEUS: Vehicle to Vehicle Communications

Walid Dabbous Christian Huitema

August 1988

1 Introduction

This report includes three parts:

- A general study of communication scenarios, application structure and application requirements,
- A study of the possible transmission control procedures for broadcast, point to point and point to multipoint communications,
- A study of the adequacy of various medium access control to the PRONET requirements: frequency of messages, large number of mobile sources, cellular or non-cellular mode...

The conclusion of the report will be structured according to the OSI layered model: for each communication layer, the recommended protocols choice will be listed.

2 Structure of the application level communications

In this section, we study various communication scenarios. we want to enlight the architecture of the application, as well as the type and quality of service required to the lower layers.

2.1 Preliminary hypothesis

Our analysis will be based on a number of requirements expressed in the PRONET report [1] and in the subsequent PRONET proposal [2].

2.1.1 General architecture principles

It appears clearly from previous documents, and discussions with M. Kemeny, that the application will have two main components:

- A *"communication system"* which will acquire informations from the external world, e.g. from nearby vehicles, queues and junctions.
- An *"expert system"* using artificial intelligence techniques which will suggest decisions to the driver based on the information acquired by the communication system.

The interface between these components can be thought of as a data base, where information on various entities and objects will be stored (see figure 1).

Indeed, this first view is over simplistic:

- The communication system will probably not merely signal that vehicle V_i is now at location L_i with speed S_i. It can also be used to signal events, e.g. a new vehicle as been identified, or intentions, e.g. vehicle V_j signals its intention to brake.
- The expert system may find conditions that require the sending of special warnings to its partners, e.g. when the driver intents to leave a queue.
- There may well be other sources of informations than the expert system and the communication network, e.g. anti-collision radars or temperature sensors.
- The data base can also be used by the communication system, e.g. in order to choose the most appropriate channel for sending the information.

In order to enlight the communication requirements, we will analyze a number of scenarios proposed in the preliminary PRONET report [1]. Note that we are making here some hypothesis on informations available from other sources. We expect that other systems will provide positional information with adequate resolution (PRO-ROAD), information about road geometry (PRO-ROAD), information about clear area ahead of the vehicle (PRO-CAR) and that each vehicle will have an unique identificator.

2.1.2 Communication scenarios

The PRONET report [1] cites a number of accident prone situations, where better communication could significantly increase the driving security. For each of these situations, a communication scenario is analyzed:

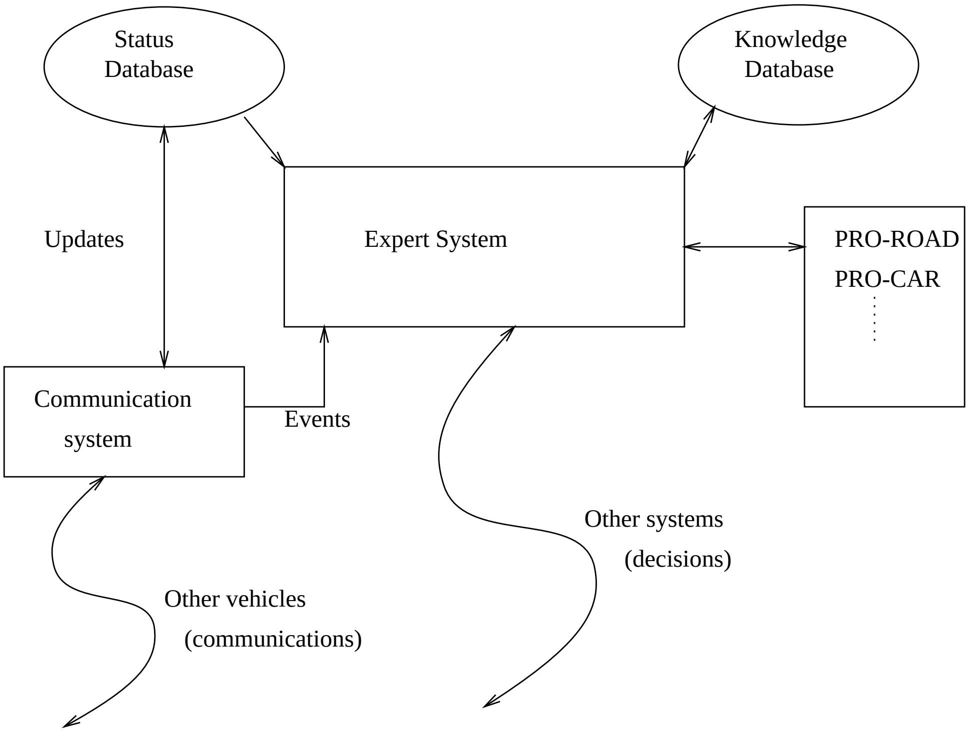


Figure 1: The PRO-NET general application architecture

- **Merging lanes – highway:**
  A vehicle is entering a highway whilst another vehicle is driving at approximately the same speed on the rightmost lane. Cooperation is required between the two vehicles – which don't "see" each other. The report suggest the exchange of approximately 2 short messages per second, during all the period.

- **Overtaking – rural road:**
  One vehicle is overtaking another on a curve road. As it does not see beyond the overtaken vehicle, it will simply announce its intention to overtake, and wait for an *"OK"* message – returned when appropriate.

- **Crossing vehicles – urban traffic:**
  Two vehicles, which don't see each other, intend to pass straight through the same intersection. By cooperating, they will safely pass the intersection with minimized longitudinal adjustment; cooperation may include 5 short messages per second.

- **Convoy driving – highway:**
  A vehicle convoy is driving at high speed on a highway; the distance between vehicles is less than 1 m. PRONET provides the organizational control of the convoy, such as planned speed changes, joining/leaving manœuvres, etc. The estimation of traffic is five short messages per second, per vehicle.

- **Accident in fog – highway:**
  An accident occurs, causing one vehicle to stop in one of the lanes of a highway. A simple transceiver is triggered on this car, to start transmitting emergency signals. The emergency signal is repeated continuously, e.g. 3 times per second. The content of the signal is for further study, as it could either be very simple – every vehicle around being notified – or quite sophisticated, e.g. giving the position of the accident.

These scenarios give us a first idea of the communication requirements.

2.1.3 Basic communication requirements

It appears from the analysis of these different scenarios that each vehicle is likely to process a relatively small number of messages per second, and that these messages are of two kinds:

- Some messages, like the "accident signal", are just sent to all vehicles "nearby". This needs also appear in the "merging lanes" and "crossing vehicles" scenarios, where the vehicles arriving to the junction must identify themselves prior to any synchronization.

- The other messages are sent to a very well identified vehicle, in order to perform a synchronization task (merging lanes, crossing vehicles), or to require and give information (overtaking, convoy driving).

This identification of the needs is essential in order to derive the requirements to the transmission layer. In particular, the frequencies of messages which are mentionned in [1] should be backed up by some detailed analysis.

2.2 A detailed scenario

The frequencies of messages quoted in 2.1.2 are derived from the early report [1]: they are seldom higher than a few hertzs. In a revised proposal [2], much higher figures are quoted, e.g. 500 Hz. Indeed, such a difference would induce drastic changes in the communication protocols design, and much higher bandwidth requirements. In order to evaluate more precisely these figures, we will choose one *"typical"* scenario, the case of merging lanes, and analyse it in detail.

2.2.1 The merging lanes scenario

The primary goal of the PROMETHEUS project is to increase driving security; a fall back is to enable driving at higher speed in secure conditions. We examine the "merging lanes" scenario, with the following hypothesis:

- The vehicle V_1 is entering a highway at high speed, e.g. 30 to 50 m/s, through a junction. It is able to recognize this situation, e.g. through informations from the PRO-ROAD system.
- One or several vehicles are driving on the rightmost lane of the highway, at similar speeds.
- At least some of these vehicles are equipped of a Prometheus system. They can communicate via PRO-NET and use PRO-ROAD and PRO-CAR facilities. In particular, they can use an anti-collision radar to determine the distance between themselves and the first vehicle in front of them.

The safe merging can be determined by the following algorithm:

1. V_1 broadcast a message, signalling its presence on the junction, and wait for responses.
2. Vehicles nearing the junction on the highway respond by their position and signal whether or not there is free space in front of them.

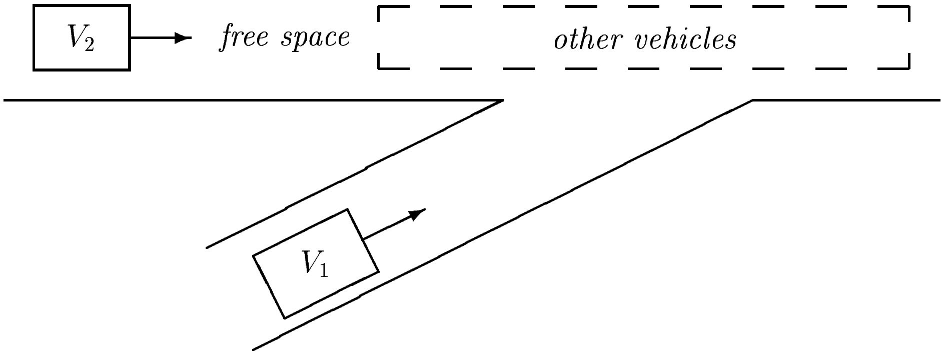


Figure 2: Situation before merging

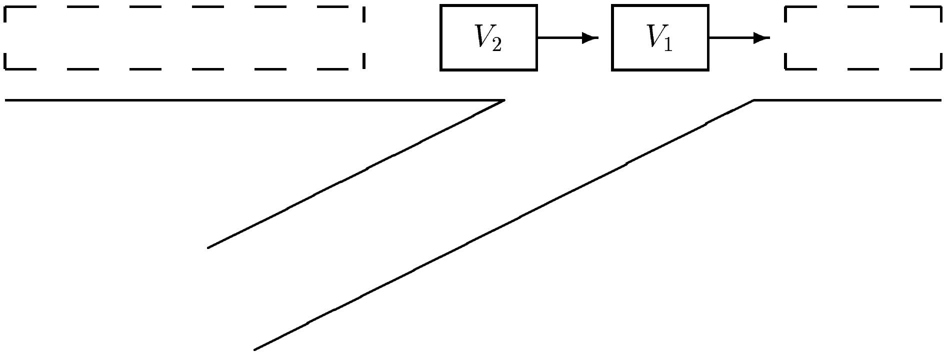


Figure 3: Situation after merging

3. If V_1 receive at least one response indicating the possibility to insert itself in front of another vehicle V_2, it proceed to step 5.

4. If after some delay V_1 does not receive a response it repeats step 1. If the junction is nearing and no information was received from the vehicles on the highway, it should reduce speed and let the driver proceed *"humanly"*. Note that if we had the guarantee that all vehicles are equipped by Prometheus, there would be no need to reduce speed...

5. A dialogue between V_1 and V_2 will convey all necessary informations, so that they can safely move from the situation of figure 2 to the situation of figure 3.

The next sections will be devoted to the analysis of this dialogue, following two possible procedures: periodic indication of position in 2.2.2 or indication of previsions in 2.2.3.

It is quite obvious that this scenario is worth further refinements, in particular to cope with the case where several vehicles are simultaneously joining the highway. The most simple case is indeed that if V_2 has already granted the free space in front of him to another vehicle, it will not particpate in any other dialogue. A possible sophistication would be for V_1 to recognize that there is enough space in front of V_2 to also accomodate a second vehicle, and to annouce to V_2 that the "length" of V_1 has suddenly increased...

2.2.2 Periodic broadcast

The periodic broadcast strategy is assumed in [2]. Each vehicle engaged in the junction will periodically broadcast a status message, containing its identification, its position and its speed.

All the vehicles will thus be able to build up a "status data base", and to adopt adequate strategies. The efficiency of this scheme is indeed very sensitive to the frequency of the updates: if the position is only approximately known, the vehicles will be obliged to compute wider margins, i.e. to maintain more "free space" between them. This will result in a general slow down of the traffic.

The figures presented in [2] assume a collision distance accuracy of only 5 cm, and a maximum speed of 50 m/s. Thus, a new position must be changed after every move of 5 cm, i.e. every millisecond if the speed is 50 m/s. This result in the transmission and reception of several 1000 messages per second; we will explain in 2.3.1 why we find this requirement excessive.

One could indeed accept wider security margins, e.g. a collision distance accuracy of 50 cm, which would divide by 10 the frequency of messages. But one can also observe that all this information is quite predictable, and devise a scheme based on the prevision of the positions, which will drastically reduce the transmission and processing requirements.

However, the "periodic broadcast" strategy has at least one big advantage: it is very robust. Even if a large proportion of the messages (e.g. 20 %) are lost due to transmission errors, the system remain secure: a new position will anyhow be received shortly after.

2.2.3 Prevision of position

Let's go back to the algorithm detailed in 2.2.1. The vehicle V_1 has been able to identify a vehicle V_2 in steps 1 to 5. Now, the two vehicles have to synchronize:

- V_1 has to adapt its speed to that of V_2, in order to guarantee that the foremost part of V_1 will be included in the free space in front of V_2, and also that the rearmost part of V_1 will be at a safe distance from the front of V_2.

- In principle, V_2 should try to maintain a constant speed. However, it may have to slow down if the vehicle in front of him slows down: V_2 must guarantee that the free space remain sufficient.
- In some conditions, V_1 may require V_2 to reduce its speed, e.g. if it cannot accelerate quickly enough in the final phases of the junction.

Let's start from step 5 from the algorithm. V_1 has identified that there is a hole in front of V_2. It must first signal to V_2 its intention, and get the agreement from V_2 which will thus commit itself to maintain enough free space. The first messages will thus be:

- From V_1 to V_2: hello, I will insert in front of you, I need X meters of free space.
- From V_2 to V_1: OK; my position is now p_2 and my speed v_2. The free space is F (in some cases, the free space may be "infinite").

At that stage, V_1 compute the date at which it will arrive to the junction. If necessary, it modify its speed, e.g. in order to aim precisely at the middle of the "free space"; one constraint is indeed that the speed at the moment of the junction should not be too different of that of V_2.

V_2 has announced a position p_2^a and a speed vector[1] v_2^b at a date t^a. It will have to compute regularly the difference between the current values and the predicted value, i.e.:

$$\delta p = p_2^a + v_2^a(t - t^a) - p_2$$

$$\delta v = v_2^a - v_2$$

If either δp becomes larger than the distance accuracy or δv becomes larger than the admissible difference between vehicle speed at the moment of the junction, a refreshing message should be sent.

In the absence of external perturbations, the necessity of sending these messages will be determined by the precision of the odometers[2]. If we assume an accuracy of 2%, a maximum acceptable δp of 50 cm and a speed of 50 m/s, we will have to send a message at most every 500 ms: we are back to the 2 messages per second which were quoted page 4. If the vehicles start synchronizing 250 m before the junction, they will exchange a total of 10 messages in 5 seconds.

---

[1] In this scenario we considered only the longitudinal component of the speed vector, in other scenarios the transversal component should also be taken into account

[2] The guaranteed precision of actual speed meters is approximatly +10%. Two problems cause this permanent incertitude: the diameter of the wheels, and the count of wheel turns in a given period — an integer number of turn fractions. The first problem can be solved by calibrating: when possible, measure the speed at which beacons are passed.

Merging lanes scenario			
Hypothesis	Speed		50 m/s
	Position accuracy		0.50 m
	Speed accuracy		1 m/s
	Speed precision		2 %
	Max. deceleration		10 m/s^2
	Frequency	Messages	Duration
Identification	5 Hz	1+n	0.2 sec
Connection phase		2	0.1 sec
Synchronization	2 Hz	10	5.0 sec
Emergency	10 Hz	50	5.0 sec
Total	2 – 10 Hz	14 – 64	5 – 10 sec
Maximum transmission delay			50 ms

Table 1: Summary of communication requirements

The worst case is probably given by a brutal slowdown of the vehicle in front of V_2. Let's suppose that for some reason, this vehicle decide to reduce its speed with a deceleration of 10 m/s^2 — certainly an extremum for the current technology. V_2 will transmit a message every time its speed has changed more than the speed accuracy. If we suppose that this accuracy is 1 m/s, a total of 50 messages will be sent every 0.1 seconds for five seconds.

All these figures are summed up in the table 1. They depend indeed of the parameters that we have mentionned, i.e. speed, speed and distance accuracy, precision of the odometer and maximum deceleration. The table also show a parameter which we have not yet computed, i.e. the maximum transmission delay, i.e. the transmission time within the network.

This delay is not so easy to evaluate. We merely choosed 50 ms as one half of the period of the maximum frequency. We will see in 2.3.3 that we cannot expect a too short response time, and that we should use some techniques in order to cope with variable delays. One very simple solution would be to carry in every position message the date at which it was issued by the sending vehicle. We will then have to devise a procedure to synchronize the clocks used by both vehicles.

2.3 Requirements and constraints

Designing an application without taking into consideration some hardware constraints could lead to an extremely expensive system. We will thus review three major constraints: processing capability, transmission data rate and response times.

2.3.1 Processing capabilities

When reviewing the "perodic broadcast" strategy in 2.2.2, we mentionned that the figure of one message per millisecond per vehicle was largely irrealistic. Indeed, one could probably devise a 100 Mbit/s network that would be able to carry such a traffic. However, we have to consider the time which would be left to process each message: if each vehicle emit a message every 1 or 2 ms, and if we have some 200 vehicles in the vicinity, each vehicle will have to receive 200 000 messages per second, i.e. the processing of each message will have to take less than 5 μs. Even if we make the hypothesis that the vehicle will only transmit their position every 10 or 20 ms, we end up with less than 100 μs to process a message.

This is indeed very irrealistic: today, high performance micro computers, based on Motorola 68020 or Intel 80386 micro processors, cost approximately the same price as a medium sized car. One may well expect that their price will drop in the following years, so that one could afford one of them per vehicle[3]; one may also expect that, for the actual price, one will get a much more powerfull device. But one may not expect to get a much more powerfull device for a much reduced price... Today, this high performance micro computers can only process a few hundred messages per second: this should be considered as a limit in the design of our application.

In practice, this forbids all approaches based on systematic broadcast of positions to all vehicles: the bulk of the traffic will have to be point to point, so that a vehicle only processes those messages which are explicitely addressed to it. In that condition, an update rate of 100 Hz could be accepted. But each vehicle may have to follow several simultaneous conversations, e.g. one with the vehicle in front and one with the vehicle behind, and the on board processing system must also perform other tasks for the PRO-ROAD and PRO-CAR applications. Thus, we should rather limit our design onto the exchange of at most a few tens of messages per second.

2.3.2 Transmission rate

Another limit imposed by the current computer technology is that of transmission rate. In 2.3.1, we have seen that requiring to process more than a few hundred messages per second would be irrealistic. This limit on processed messages does not imply that the total number of messages on the network should be less than a few hundreds per second. Almost all local area network attachments include some hardware filtering, so that a station will ignore all the paquets which are not directly sent to it, either by broadcast or by point to point addressing.

---

[3] The prices of vehicles are not expected to drop in the same proportions

However, another limit exists, i.e. the speed at which messages can be stored into memory. Most local area network VLSI controllers have only a limited amount of internal buffering — a few bytes or words. They rely on accessing the system memory through some "Direct Memory Access" device. This is only possible if the access rate through the DMA is significantly larger than the transmission rate from the network. Currently, the input channels for micro-computer systems have a capacity varying between 1 Mbyte/s for a PC and 5 Mbyte/s for a 68020 or 80386 based work stations; this limits the speed of local area networks to approximately 1 Mbit/s for PCs (e.g. StarLan, Appletalk), 10 to perhaps 16 Mbit/s for work stations (Ethernet, second generation token ring).

One could indeed bypass this limit by implementing fast double access memories. This solution is part of the design of AMD "Supernet" implementation of the 100 Mbit/s FDDI token ring, which include 256 Kbytes of extra fast double access memory, used for input and output FIFO. However, the price of such attachments can become quite expensives. Figures of several 1000 US$ are rumored...

If we want to design an inexpensive on board network, we should try to stay within the actual 10 Mbit/s range. The figures from table 1 show an average frequency of 2 Hz for a given communication; if we assume a message size of 1000 bits, a data rate of 10 Mbit/s would allow a maximum capacity of 5000 simultaneous conversations at average rate. Indeed, we should not expect a 100 % utilisation of the channel, as some overhead will be induced by the lower layer protocols: 50 % would appear realistic. Also, one could assume that some of the conversations will use the maximum frequency of 10 Hz also quoted in table 1: if we accept to see a maximum proportion of 20 % for the emergency rate, we get an acceptable number of conversations of:

$$\frac{50\,\%.(10\,Mbit/s\ /\ 1000\,bits)}{80\,\%.2\,Hz + 20\,\%.10\,Hz} = \frac{5000}{1.6+2} \simeq 1388$$

This number is compatible with the hypothesis made in [2], that approximately 250 vehicles would be present in a 500 m diameter "communication cell". Actually, this computation shows that we could even accept a data rate lower than 10 Mbit/s; for 250 conversations, a reverse computation:

$$\frac{250\,conversation.(1.6+2\,Hz).1000\,bits}{50\,\%} = minimal\ bandwidth$$

shows a throughput of only 1.8 Mbit/s. In practice, the bandwidth allocated to our system will have to be somewhere between 2 and 10 Mbit/s; it will be determined by the VLSI available – we could perhaps reuse some LAN designs – and also by the available radio frequencies.

If we had to rework the figures of message traffic, or those of conversations, and if we were to end up with much larger requirements, we would have one

Action	Delay (μs)	
Emission protocol	t_{pe}	250
Media access	t_{mac}	2000
Emission time:		
2 Mbit/s, 1000 bits	t_e	500
Transmission delay: *500 m*	t_d	2
Reception interrupt	t_{ir}	2000
Total time	t_t	4752

Table 2: Components of the transmission delay

possibility, that of using several distinct communication channels, e.g. one radio channel for broadcast and another one for point to point communication, so that the sum of the two bit rates remains lower than the capacity of the I/O channels of the processors.

2.3.3 Response times and clock synchronization

We had quoted in table 1 a maximum "response time" of 50 ms. Indeed, the simple addition of the transmission times t_e, with the conservative hypothesis of a 2 Mbit/s channel, and of the transmission delay t_d with a maximum distance between vehicles of 500 m, yields a much lower figure:

$$t_e = \frac{1000\,bits}{2\,Mbit/s} = 500\mu s$$

$$t_d = \frac{500\,m}{3.10^8 m/s} \simeq 1.33\mu s$$

$$t_e + t_d < 502\mu s$$

However, the total emission time will also contain other factors, e.g. processing the emission protocol in the sending machine t_{pe}, and acknowledging the reception interrupt in the receiving machine t_{ir}. Also, the media access protocol may induce some queuing. From past experience, we can assume the order of magnitude shown in table 2.

This delay of approximately 5 ms is still much shorter than the 50 ms that we mentionned page 9. However, one major source of delays is still missing, i.e. the retransmissions. The transport control protocol described in section 3.3 is based on timers: if an acknowledgement does not arrive within a given delay, the message is retransmitted. This process can be repeated several times if successive errors also cause the loss of the repeated packets.

The probability that a given packet will have to wait n retransmissions depend of the packet error rate p_e:

$$p(n) = (1 - p_e).p_e^n$$

Hypothesis	p_e	20 %
	t_t	5 ms
	timer	10 ms

Retransmissions		Total
Number	Probability (%)	Delay (ms)
0	80.000	5
1	16.000	15
2	3.200	25
3	0.640	35
4	0.128	45
5	0.026	55
6	0.005	65
≥ 7	0.001	≥ 75

Table 3: Distribution of delays after retransmissions

Most systems are programmed with conservatively large timer values, as they must cope with a large number of environments: in some concatenations of local and wide area networks, the transmission delays may well exceed a few seconds. In our case, we will have on the contrary to use very short timers, in order to proceed rapidly to the retransmissions: for example, if we have computed a normal transmission time of 5 ms, we can assume that the acknowledgements will return within 10 ms.

If we assume a rather large packet error rate of 20 %, and a 10 ms timer, we obtain the distribution of the final transmission delays figured in table 3. This distribution shows that it is important to maintain a very good packet error rate on the transmission channel, in order to have stable transmission delays. It also shows that, under some circumstances, the delay can be much larger than 5 ms.

We have mentionned in page 9 that one way to cope with these variable delays was to include in each position packet the *date* at which it was sent. But, for a number of reasons, the clocks on two vehicles have few chances to be synchronized: one is Einstein's theory of relativity, and another one is the precision of the clocks, seldom better than 10^{-7}. Thus, in order to be useful, the dates have to be aligned: V_1 must keep in memory that the clock of V_2 is δt_{1-2} ms in advance of its own, for the duration of the communication. The initial value of δt_{1-2} can easily be computed during the first exchange of the conversation; it could then be tracked as the lesser value (algebric) of the difference between the date t_2^i in the i^{th} message from V_2 and the value t_1^i of the clock of V_1 when this message is received: if the clocks remain stable during the whole conversation, the precision of the adjustment will be approximately

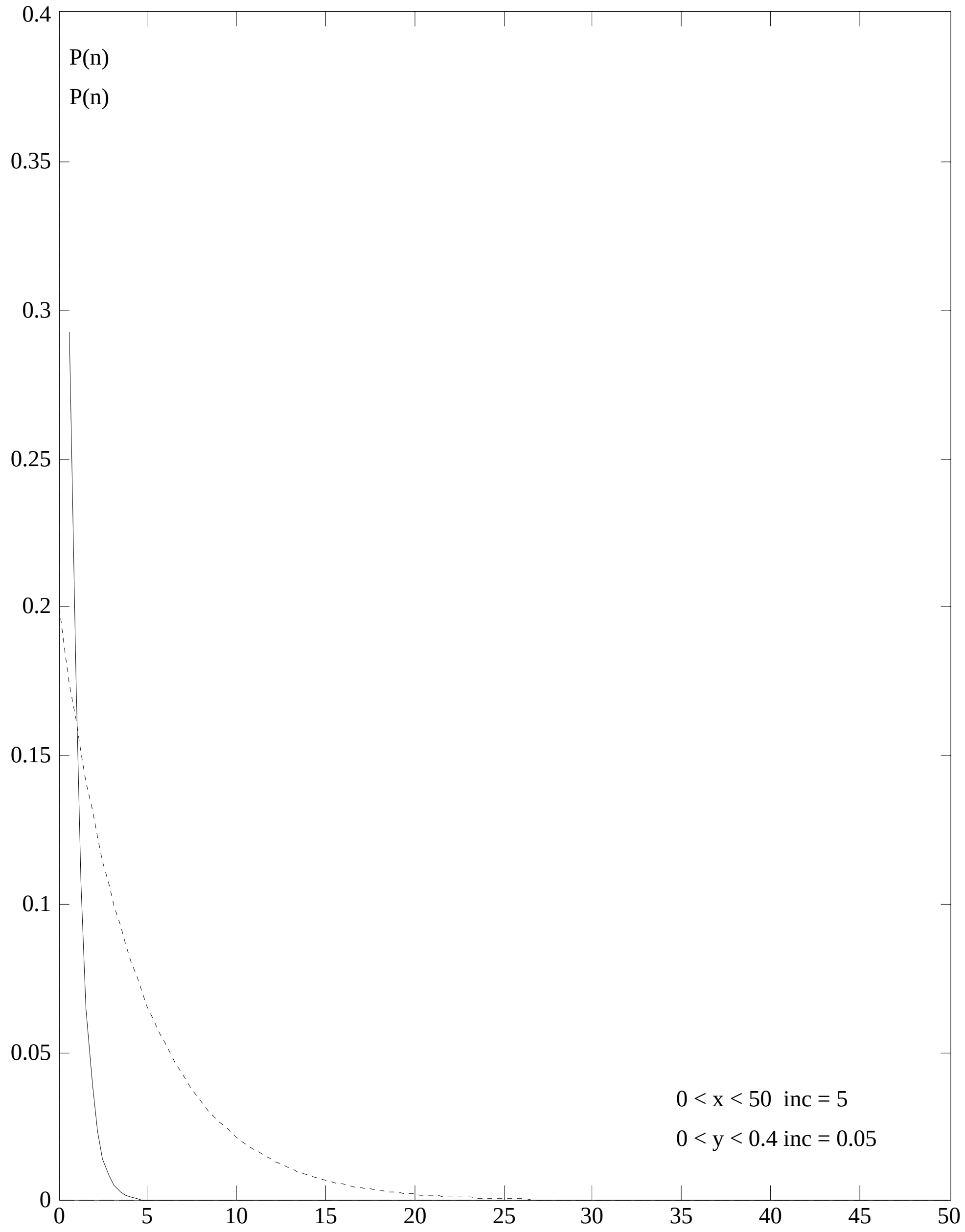


Figure 4: Probability of having n retransmissions versus n for two different values of p_e. Notice the rapid decrease when $p_e = 20\%$ comparing to the slope for $p_e = 90\%$.

t_t, i.e. the minimum time necessary to transmit and receive a message, 1 to 5 ms in our example. If the speed is 50 m/s, the position accuracy will vary from 1 to 5 cm.

Note that if the conversation were to last for a very long time, e.g. if a queue of vehicles remain stable for several hundreds of kilometers, more sophisticated procedures would be necessary: one should compute two parameters for the initial difference and the slope of variation.

2.4 An application level summary

The previous analysis has shown that, in order to build a realistic network, we have to limitate the number of messages in general and, most of all, the number of broadcast messages. Hopefully, the review of scenarios in 2.1.2 shows that broadcast messages will only be sent in exceptional conditions: *ALARM* in case of accident, *HELLO* when PRO-ROAD signals that one is nearing a junction or when PRO-CAR signals the presence of an unidentified car ahead. The *HELLO* packets will normally trigger a handshaking procedure, and the remaining traffic will then be sent only to the identified partners: they will not overload the processors of the other vehicles.

This two phases communication structure will make the evolution of the system much easier. Broadcast messages must have a very rigid structure, as all receivers must be able to understand them. The handshaking procedure, on the contrary, may well include the negotiation of a "version number": more modern systems will still be able to communicate with the early ones, loosing only some of the communication capabilities.

The detailed analysis of the "merging lanes" scenario showed that all figures quoted in 2.1.2 must be backed with detailed descriptions of the driving procedures. If the assumptions are right, which is not contradicted by the analysis in 2.2, we aim at a network carrying a few Megabits per second, where each vehicle only processes a few tens of messages per second.

A secondary benefit of the detailed analysis is to clarify the relation between the different components of PRO-CAR, PRO-ROAD and PRO-NET:

- An expert system, using informations from PRO-ROAD and PRO-CAR, HELLO or ALARM received from PRO-NET, and also informed of the driver intention, will detect the need to activate a particular "scenario".
- To each scenario correspond a recorded procedure: for example, to the merging lane scenario correspond the procedure described in 2.2. In most cases, it will include some broadcasting of information, followed by the identification of a "partner", to which all other messages will be sent.
- This procedure will consist of scanning local variables, e.g. the distance to the vehicle in front, sending messages and suggesting actions when

conditions are met or when messages are received from the “partner” vehicle.

The requirements to the communication system will include at least two types of service: broadcast messages, which will not be “secured” by acknowledgments, and vehicle to vehicle connections, where there should be a guarantee that “lost” messages will be retransmitted within a very short delay. The quality of service requirements are that the capacity should be at least a few megabits per second, and that the packet loss rate should be kept low, so that few retransmissions are encompassed.

3 Control of the communications

The previous analysis showed that the information flow was either broadcast or sent to identified receivers. In some particular cases, more than two recipients are interested in a communication: we call “conference” this particular case of point to multipoint communication where each participant listen and talk to all others. In this section we will analyze the required transmission control procedures and propose adequate protocols for both the network and transport levels. This analysis is based on the OSI reference model [3].

3.1 Network protocol

Within the OSI reference model two Network services are considered, the Connection Oriented Network Service [4] and the Connection Less Network Service [5]. The CONS is obviously inadequate due to the varying network topology. The connectionless network service is adequate as messages are typically short and we may need to broadcast to all the nearby vehicles. The connectionless network protocol mentionned in the OSI standards is known as ISO-IP [5] due to the similarities with the DARPA IP protocol. The main primitives provided by this network layer protocol are addressing, routing and segmenting.

3.1.1 Addressing

The network addresses define the interface between the nework and the transport layers. In the OSI model these addresses are used to identify NSAPs or Network Service Access Points. An NSAP is an abstract information identifying a Network Entity. In our case NSAPs will characterize vehicles. Each vehicle will have a *network address* uniquely identifying it. Many schemes can be used for the network addressing: flat and hierarchical addressing. With the first scheme, the vehicles should have a complete view of the newtwork while with the hierarchical addressing (as for NSAPs) a vehicle will only know the routes to the “nearby” vehicles and the routes to other domains or cells. The second scheme can be used for cellular mode communications where the NSAP address includes a field identifying the domain to which belongs the destination vehicle, and another field to address the vehicle within this domain. The hiererchical structure of NSAP addresses allow to achieve economical routing between different domains. If a vehicle wants to communicate with another one, the transmitted packet should include an encoding of the source and destination NSAPs. We should also have an NSAP for the broadcast address.

3.1.2 Routing

Within a given domain several channels instead of one may be used to improve performance (see Sec 4). For example, we can have one infra-red point to point “bumper to bumper” and one broadcast radio channel. Another assumption consists of having several radio channels. The role of the routing primitive is then to determine which channel is to be used to communicate with a given vehicle. Each vehicle will maintain a routing table giving the channel associated with each of the other destinations. This table is dynamically updated e.g. upon reception of a HELLO packet: in this case a new entry is added. The dynamic routing i.e. start of communication on one channel and continuation on another is also allowed. It is one of the advantages of the connectionless model.

3.1.3 Segmentation

This primitive is used for relaying mass size information in networks. In fact, if we restrict ourselves to some limitations we can implement the *inactive network layer* as mentionned in the ISO-IP standard in order to obtain an almost null overhead. Otherwise, if the full addressing and routing functionalities should be present and if the medium supports large packets so that segmentation is not required we may implement the *Non-segmenting* subset of the full protocol.

3.2 Multipoint transport transmissions

This mode of communication allow to set up “conference” calls between vehicles. It is used in satellite communications e.g. NADIR project with point to multipoint transmissions. Multiple transport connections are more difficult to control: how to pass the right to transmit from one vehicle to another, how to collect and order messages incoming from different vehicles. For these reasons we prefer to use several point-to-point connections instead of a conference call.

3.3 Transport Protocol for point to point transmission

For the transport layer the two communication modes are required: one connectionless transport service for ALARM and HELLO signals which should reach their destination as fast as possible, and another connection oriented transport service used for reliable data transfer. In fact, packets may be lost or damaged due to transmission errors. The use of error detection and recovery mechanisms is then required. The dialogue between two vehicles will include three phases: a connection establishment between the two communicating entities, then data can be exchanged reliabily and finally the connection is cleared at the end of the dialogue.

One of the most famous standard transport protocol is the OSI transport protocol class 4 [6]. This protocol gives the required point to point connection oriented functionalities by the use of error detection and recovery primitives. It also provides a set of usefull procedure like multiplexing, and expedited data units. We can resume its functionalities as follows:

1. Use of checksum mechanism to detect and then discard damaged data units.
2. If packets are not acknowledged within a specified time delay (T1 timer) a retransmission is performed. This requires the numbering of data and acknowledgment packets. The protocol ensures the sequenced delievery of the transmitted packets.
3. Multiple transport connections may be established between two vehicles using the same network connection. This is usefull if vehicles have to negotiate different applications in the same time.
4. The use of expedited data units allows the transmission of two traffic classes. An expedited data unit will bypass the normal queuing delays in the vehicles: it is processed as soon as it is received.

However, the protocol procedures need some optimization e.g. the use of an adaptive retransmission timer in order to avoid unecessary retransmissions due to a late received acknowledgment.

4 Media access protocols

Two alternatives exist concerning the choice of the transmission media: either one or several transmission channels. The first approach seeks for simplicity in the transceivers; informations are transmitted on the same broadcast channel. All the vehicles listen and transmit on this channel. Only the concerned vehicles copy the message and process it. This approach is limited specially when the number of vehicles increases. In this case overlapping may occur and the medium access protocol operates with a very low efficiency. The other alternative is to have multiple channels e.g. a broadcast channel for *ALARM* and *HELLO* messages and several others which may be either point to point or multiaccess shared channels. Vehicles use the broadcast channel to identify the other communicating vehicles and to negotiate a channel to continue the dialogue.

4.1 How to share a communication channel

A medium access protocol is needed for a multiaccess network. Such protocols are designed to address the problem of how to share a common broadcast channel. Many techniques exist and some have become standards for local area networks, e.g. the IEEE 802 medium access and logical link control protocols [7]. The Medium Access Protocol (MAP) is either *controlled* or *random* access depending if only one vehicle is allowed to transmit at a given time or more. The first case requires an explicit or implicit logical order for the vehicles having the right to transmit, usually acheived by a Token Passing protocol. On the other hand the contention protocols like ALOHA and CSMA are satisfactory when systems are not overloaded, due to the random access mechanism. The analysis of the adaptivity of each protocol for vehicle to vehicle communication follows.

4.1.1 Token Passing

The token passing is an ANSI/IEEE Standard (802.4) [7] for the medium access control sublayer. This technique consits of predetermined resolution of the medium access problem by allowing only the station having the *token* to transmit. When a station receives the token it is granted the control of the medium for a specified time. The station may either send a message within the time limits or pass the token to the *next* station, with the last member of the sequence followed by the first. Each station knows the identity of the stations preceding and following it in the *logical ring*. This technique is usefull when stations have a high offered load to transmit, but is not adequate when the network topology is not known a priori, as in our case. In addition, the network initialization and reconfiguration (addition or deletion of a station) phases needed for the proper operation of the token ring may not lead to an operating configuration. The token loss is hard to detect and leads to uncertain situations. Furthermore, a station having an ALARM message can be delayed until it receives the token. The token protocol is then not adequate for the vehicle communications.

4.1.2 ALOHA

ALOHA was one of the earliest contention based techniques [8]. This protocol is applicable to any transmission media shared by uncoordinating users (a free-for-all protocol). Whenever a vehicle has a message to send it does so. Then the vehicle "listens" for an amount of time equal to an estimate of the maximum possible round-trip propagation delay on the media (twice the time it takes to send a message between the two most widely seperated vehicles.) If the vehicle finds that the message was correctly received by the destination by hearing no other vehicle transmitting within the vulnerable period, fine; otherwise a conflict was detected, the vehicle retransmit the message some time later. ALOHA is adequate when the number of vehicles is large, with a relatively low and "bursty" traffic [9], [10]. It does not require a priori knowledge of the number of the communicating vehicles (flexibility to join/leave "active zones" or cells). There is also the possibility to use multiple frequency bands for different communications in order to prevent collisions.

This protocol seems to respond to the application needs but its uncontrolled access mechanism reduces the efficiency of the channel to a minimum value. The theoretical value of 18% is reached only with an offered load of 50% (see figure 5). The operation at this maximum efficiency results in very high collision rate (65 %) which translates due to retransmissions procedures into higher delays, often unacceptable (see table 3).

4.1.3 CSMA

Another contention based technique consists of listening the media before transmitting in order to avoid collisions when another vehicle is talking and then enhance the usefull channel capacity and reduces the collision rate. This technique is known as the CSMA (Carrier Sense Multiple Access). If a vehicle detects the "carrier" i.e. another vehicle using the channel it backs off some period of time and tries again. It may happen that two or more vehicles attempt to transmit within a time interval smaller than the round-trip propagation delay. In this case there will be a collision. The precise merit on ALOHA is the increase of the channel utilisation based on sensing the channel before transmitting [11]. So the performance of CSMA depends on ability to detect that the channel is busy. This detection is not always easy as we are dealing with moving vehicles: Vehicle V_i may belong to the intersection of two active zones or cells, and due to the "multiple path" problem, two vehicles each belonging to a different active zone may transmit in the same time to V_i. As the two other vehicles are out-of-sight the later one cannot detect that the first has started transmission: a collision occurs and V_i does not receive neither of the messages correctly. This problem was studied in detail in [12]. The proposed solution requires the existance of a central station in

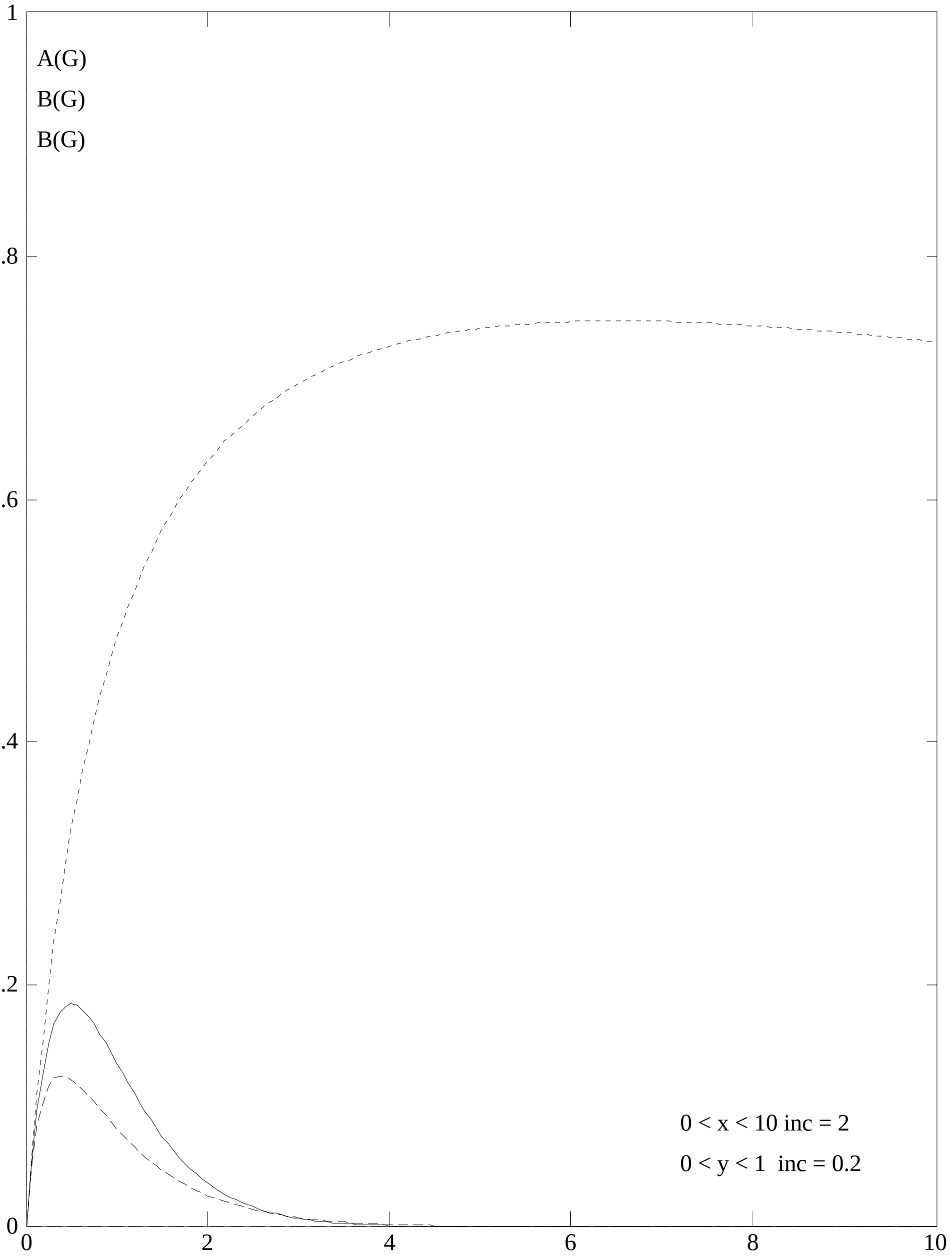


Figure 5: Effective throughput S versus offered load G for ALOHA (continuous line) and CSMA (dashed lines) protocols. The best throughput is with CSMA for $a = 0.02$ while for $a = 1.2$ ALOHA exhibits higher throughput.

line-of-sight of all other stations. Another example could be the directional "bumper"-to-"bumper" infra-red channel: two vehicles trying to communicate on the infra-red channel with a third one cannot detect the "collision" of the two communications.

The efficiency of the CSMA protocol depends on the ratio of the round-trip propagation delay and the packet transmission time:

$$a = \frac{\tau}{T} = \frac{\tau.C}{L}$$

where C is the capacity of the channel in bits/s and L the packet size in bits. The improvement gained by sensing the channel is of less importance if this information is "old" with respect to the packet transmission time i.e. $\tau \approx T$. We can then derive a bound on the capacity for a fixed packet size and round-trip propagation delay in order to maintain a given efficiency. Besides, for any value of a the tradeoff between the throughput of the channel and the offered load is given by:

$$S = \frac{G.e^{-aG}}{G + 2aG + e^{-aG}}$$

This equation, proved in [12], shows that:

$$\lim_{a \to 0} S = \frac{G}{G+1}$$

i.e. if the propagation delay is negligible the effective throughput is equal to the maximum capacity of the channel at high offered load.

As a increases the maximum acheivable throughput decreases, and may even become lower than the efficiency of ALOHA. It is not then usefull to use CSMA unless a is much lower than 1.

The value $a = 0.02$ corresponds to a maximum throughput of aproximatly 75% of the capacity of the channel. For 1000 bits packets and 500 m active zone radius the maximum capacity is given by:

$$C = \frac{L.a}{\tau} = \frac{20}{2 \times 500/3.10^8} \simeq 6\, Mbit/s$$

and the effective throughput is $S = 4.5\, Mbit/s$. Higher values of C, e.g. $C = 40\, Mbit/s$, result in $a = 0.13$ and $S = 18\, Mbit/sec$ (see table 4).

This analysis shows that with CSMA and for a fixed packet size and propagation delay the higher are the transmission speeds the lower is the efficiency. It means that if we want to use a single high speed channel e.g. $C = 100$ Mbits/s the maximum efficiency will be in the order of 30% i.e. a maximum acheivable throughput $S = 30$ Mbits/s. An optimal value of C could be around 10 to 20 Mbits/s with a throughput of 6 to 11 Mbits/s.

		C					
		2	4	10	20	40	100
G	0.1	0.09	0.09	0.09	0.089	0.088	0.085
	0.2	0.166	0.165	0.164	0.162	0.158	0.147
	0.4	0.284	0.283	0.279	0.272	0.260	0.227
	1	0.5	0.49	0.475	0.452	0.4	0.3
	2	0.655	0.643	0.6	0.557	0.46	0.26
	4	0.774	0.75	0.68	0.578	0.41	0.15
	10	0.845	0.785	0.629	0.433	0.2	0.02

Table 4: Efficiency of CSMA protocol for a given transmission capacity C Mbits/s and an offered load G. Values are computed for 1000 bits packets, the maximum distance between two station beeing of 500 m.

4.1.4 Improving CSMA

The *Carrier Sense Multiple Access with Collision Detection* technique (CSMA-CD) is the most commonly used for bus topologies; it is also referred to as "listen while talk". This technique is an ANSI/IEEE Standard (802.3), its baseband version was developed by Xerox as part of the Ethernet Local network. CSMA-CD is a refinement of the CSMA protocol attempting to overcome one glaring inefficiency of CSMA. With CSMA when two messages collide the channel remains unusable for the duration of transmission of both messages. If packets are long compared to the "bit length" of the channel the amount of wasted capacity is considerable. This waste is reduced if stations continue to sense the channel while they are transmitting: if a collision is detected during transmission, stop transmitting the collided messages, then transmit a brief signal to inform stations that a collision had occured. Then the station wait a random time before transmitting again (by sensing the channel before as in CSMA).

CSMA-CD is indeed more efficient than CSMA, in terms of maximum acheivable throughput and undetected collisions. But it supposes that the collisions can be detected easily. In the case of mobil vehicles with radio channel this task is quite impossible: when a station is transmitting, the radiated power saturates the local receiver, and hence signals from other transmitting stations will be too weak to be detected. For example if the vehicle emission power is in the order of tens of kilowatts, the transmitting vehicle will receive in the case of a collision the sum of two signals: its own signal and the collided signal. If we consider that the local antenna is 10 cm far from the emitter and that the other transmitting vehicle is 10 m away its signal will be received with a attenuation of -40 dB. It cannot be detected by the saturated local receiver.

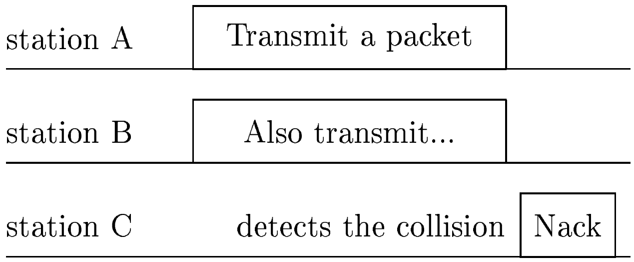


Figure 6: MAC level acknowledgments

There is probably no much way to improve the "channel utilization" ratio of the CSMA protocol. However, there may be one possibility to improve its loss rate, by including some form of acknowledgements in the medium access protocol, as indicated on figure 6. The stations A and B which are transmitting the colliding packets have no way to detect it, but the receiving stations can do it quite easily. They can, in case of collision, send a *negative acknowledgement* signal (NACK), which can be very simple, e.g. a jamming. Thus, the sending algorithm becomes:

1. Wait that the channel becomes available.
2. Emit the packet.
3. Listen the channel for a delay Δt after the end of the emission. Δt must be larger than the *diameter* of the channel, that is, the maximum delay between two stations.
4. If a *negative acknowledgement* is received, enter the *retransmission* algorithm.
5. If no NACK was received, assumed that the packet has been correctly transmitted.

Indeed, the existence of such a scheme does not guarantee that all recipients will correctly receive the packet. However, it reduces the frequency of undetected collisions, which will be retransmitted after a much shorter delay than the 10 ms quoted page 13.

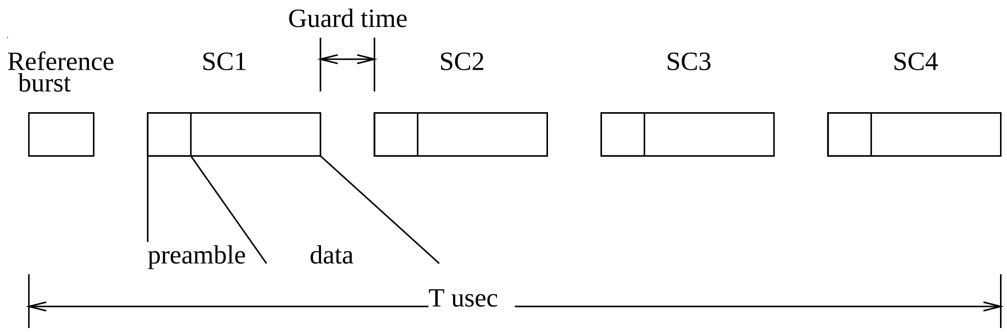
The retransmission algorithm can be derived from the linear or exponential "back-off" procedures used for CSMA-CD networks: prior to sending, each station waits a random delay dt; the average value of dt increases linearly or exponentially after each collision. One could also use a smarter *stack* or *tree* variant. The stack algorithms were studied with some detail in [13], [14]. This protocol is a "devide and conquer" algorithm; when users collide, they are seperate recursively according to some randomization procedure into two groups. Users of the first group attempt retransmission just after the collision, while those of the second group wait until the first group has resolved its collisions before they start transmitting again. The sequence of splittings may be represented as a tree, whence the name of *tree algorithm.* Each of the active users (i.e. having a message to send) determines if it has the right to attempt transmission by testing its position in a virtual stack. If it had never collided a user is in level 0. After a collision all users of level $i > 0$ change to level $i + 1$, while those of level 0 split themselves into two groups; one remains at level 0 while the other is pushed into level 1. It is only when the level 0 contains one user that the next attempt will be successfull. This control reduces the collision probabilty and then the expected time for collision resolution [15]. The implementation requires a special purpose chip in order to run the protocol efficiently.

4.2 Channels allocation policies

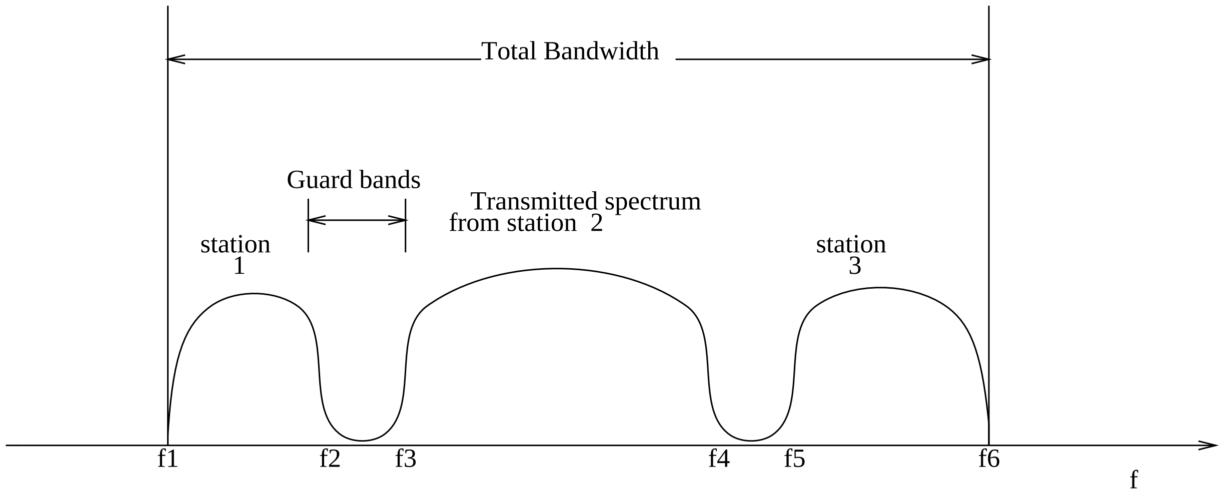
When multiple channels are used for inter vehicles communications a capacity allocation task is to be performed. In some cases the allocation is carried out by a centralized control station as in TDMA. The other case is the multiplexing in frequency domain or FDMA.

4.2.1 TDMA

With Time Division Multiple Access time is slotted and each channel can be assigned a certain number of slots. Here, each of the repetetive slots is a “subchannel” and is independent of the other subchannels (see figure 7). For example a form of switching can be achieved by including an address field. Thus a number of stations could read the data in each slot looking for data addressed to them. Broadcast messages can be sent on the diffrent subchannels. Vehicles may put a burst of data in their assigned time slot in a global frame. Each vehicle should know the transmission and reception slots corresponding to each channel. The data slots are seperated by *guard times* to ensure that there is no overlap, resulting in a certain “bandwidth” waste. To synchronize the communicating vehicles each frame begins with a reference burst provided by a fixed station. The absence of such fixed reference results in synchronization loss. Furthermore, if the number of subchannels increases the total frame length would become very large resulting in synchronization errors. The fixed assignment of slots to different channels reduces the global efficiency as vehicles are circulating countinously.



TDMA Frame



FDMA Frequency plan

Figure 7: Figure of a TDMA frame and of FDMA frequency plane

4.2.2 FDMA

In Frequency Division Multiple Access the total bandwidth is divided into channels. Each of these channels can be, in turn, shared by multiple stations (see figure 7). Each station is assigned a different carrier frequency on a fixed basis. Actually, this causes a problem of how to allocate frequency bands. To overcome this problem, and hence increase the efficiency a DAMA like technique can be used with FDMA. With Demand Assignment Multiple Access the set of subchannels is treated as a pool of available links. To establish a full-duplex link between two vehicles, the choice of a subchannel is negotiated on a common broadcast channel. Then a broadcast message is sent to inform the near-by vehicles that this chosen subchannel is reserved. For example for the merging lanes scenario vehicles can listen on the broadcast channel for ALARM and HELLO messages, negotiate another frequency band to establish the full-duplex connection for the exchange of information needed for the traversal of the junction. This choice seems to be the most adequate for the vehicles communication. However, the number of subchannels is limited by the *guard bands* necessary to avoid overlap of the transmitted spectrums. FDMA suffers also from noise and interference reducing the allowable spectrum for modulation. Finally as FDM components are onerous cost considerations should also be taken into account.

4.2.3 Spread spectrum

A third form of channel division is given by the *spread spectrum* method, or *Code Division Multiple Access* (CDMA). If we suppose that a global bandwith is available, it can be shared either by allocating all the bandwith to one station for a given time slot (TDMA, CSMA), or by allocating a share of the frequencies to one station all the time (FDMA), or by allocating a varying number of the frequencies to one station for very small time slots.

Basically, with CDMA, one station S emits the same signal on multiple channels: the precise channel pattern is the “signature” of the station. The receiver apply the signature to these multiple channels. On each channel, they will receive the signal from S and some noise – the noise may include emission from other stations. By summing up the reception from the various channels, the noise will average, and they will be able to decode the signal.

The advantage of CDMA is indeed its very good noise immunity. If TDMA is used, a complete time slot can be made useless if some noise occurs during the TDMA synchronization slots; if FDMA is used, one station can be excluded from the network if a single frequency is jammed. With CDMA, the noise will have to be spread over the whole spectrum of frequencies for a very long period in order to jam the signal: this would require much more energy. Another advantage of CDMA is its discretion: in order to receive a signal, a station must be know the “signature” of the emitter. These advantages make CDMA a very good candidate for military networks, which ennemies should not be easily able to listen or jam; it also makes it a good candidate for high frequency radio networks, which can be subject to high level of noises, e.g. in urban areas.

The usage of CDMA will typically lead to a rather low bandwidth efficiency e.g. $1/10^{th}$ of that of TDMA: in order to average the signal out of the noise, one should obviously send it in a very redundant fashion. This makes it mostly adequate for very high frequencies, e.g. in the centimetric or millimetric wavelength, where some shares of the bandwidth could be spoiled. The noise characteristics of these frequency bans should be studied in order to optimize frequency bands allocation.

The implementation of the CDMA depends of the particular signature which is choosen. In some systems, the emitter “jumps”from one frequency to the other at very short intervals; the receivers should follow the same frequency sequence in real time, which leads to severe synchronization requirements. In [16] a digital approach is proposed. Instead of “jumping frequencies”, the emitter emits on the base channel the same signal at the instants $t + d_1, t + d_2, ...t + d_n$; the receiver feeds the incoming signal into n delay lines of length $d_1, d_2, ...d_n$ and sums the outcome of these delay lines. In [16] the delays are provided by optical fibers of adequate lengths, leading to an “all optical” design – the authors are aiming at Gbit/s channels. For lower channel rates, digital shift registers could probably be used.

The CDMA channels could be used in much the same way as FDMA: a single signature would be used for the broadcast channel by all the stations, and a particular one for each point to point channel.

4.3 Medium access recommendations

From the above analysis we find that for the cellular mode the most adequate is the use of multiple channels each for a given cell, provided that these subchannels are not overloaded by external traffic. The non-cellular solution is simpler to implement but is more restrictive in terms of channel transmission capacity. If a single channel solution is adopted CSMA with stack algotithm for collision resolution is the best protocol to be used. When multiple channels can be supported FDMA with dynamic subchannels allocation allow to share the medium resources the most efficiently; Spread spectrum techniques are more expensive to implement but could provide a smaller noise sensitivity.

5 Conclusion

This preliminary report does certainly not provide a full analysis of all the levels of protocols of the Prometheus application. However, after this analysis, we have now a much clearer view of the various studies which have to be conducted. We will sort these tasks according to the various layers of the OSI model.

Physical and Link layers: The architecture of these layers will follow the model developed by IEEE for local area networks [7].

We have seen page 11 that the application could probably run with a bandwitdh of 1.8 Mbit/s, and page 22 that the CSMA protocol could be used to manage efficiently radio channels with a 500 m radius and a data rate of 6 Mbit/s, providing a throughput of 4.5 Mbit/s.

The next step is indeed to analyse all the components which are needed to set up such a channel. Several choices are still possible:

- use “standard” Ethernet VLSI, replacing the standard transceiver for coaxial cable with a radio equipment which would provide the same interface – without detecting collisions. The data rate of the channel will have to be 10 Mbit/s.
- design an ad-hoc component implementing the CSMA protocol at a data-rate of 4 to 6 Mbit/s.
- design an ad-hoc component which will implement an enhanced CSMA protocol, with collision detection and stack-based collision resolution, as described page 24

This last alternative should be sustained by a detailed modelling of the effect of the inclusion of the acknowledgments.

Few studies are necessary for the logical link control protocols. One could well adopt the solutions recommended by IEEE and ISO.

Network layer: Obviously, some connectionless network service is adequate. If we adopt a solution were the communications uses a mix of broadcast and point to point channels, the network layer will include a routing function: the choice of the communication channel suitable for a given destination.

Transport layer: Two services, one for broadcast, the other one for point to point, are necessary. The broadcast service will be based on a *connection-less* transport service; the point to point service will be the standard *connection oriented* transport service.

The protocol for providing the connectionless service is very simple. It will not include any procedures for error recovery [17].

The connection oriented transport protocol is the class 4 of the ISO transport protocol. Implementations will have to be optimized in order to provide the very short response time requested by the application (page 13).

Session layer: The session layer optionally provide services for organizing the dialogue (turn control) and for check-pointing (synchronization points). Neither of these services appears to be requested in our scenarios. The PRO-NET applications could very well use a minimal session layer, with no other functionnalities than connection management and full-duplex data exchange.

Presentation layer: All the messages used by the application will have to be described using the standard *"abstract syntax notation 1"* (ASN.1). The type of encoding which will be used during the transmissions remains debatable.

Application layer: In this report, we have analysed only one scenario in detail. The same detailed analysis will have to be conducted for all scenarios.

References

[1] *PRO-NET.* Themes for Research Collaboration with E & S Industries. Second PROMETHEUS-Symposium Oct. 1987.

[2] *A PRO-NET PROPOSAL for WG6 Concept Group.* Renault France, July 1988.

[3] CCITT, *Data Communication Networks. Open Systems Interconnection (OSI). System Description Techniques. Recommendations* X.200-X.250, Red Book, Vol.VIII, Fascicle VIII.5, CCITT, Geneva, 1985.

[4] International Organization for Standardization. *Network Service Definition* ISO/DIS 8348, 1985.

[5] International Organization for Standardization. *Protocol for providing the connectionless-mode network service,* ISO/DIS 8473, 1986.

[6] International Organization for Standardization. *Connection Oriented Transport Protocol Specification,* ISO 8073, 1986.

[7] The Institute of Electrical and Electronics Engineers. *American National Standard ANSI/IEEE Std. 802.2, 802.3, 802.4, 802.5 –1985.*

[8] Abramson, N. *The ALOHA system-Another Alternative for Computer Communications.* Proceedings, Fall Joint Computer Conference, 1970.

[9] Kleinrock, L. *Queuing systems, VOL II: Computer Applications.* New York: Wiley, 1976.

[10] Schwartz, M. *Computer-Communication Network Design and Analysis.* Englewood Cliffs, N.J.: Prentice-Hall, 1977.

[11] Tobagi F.A. and Hunt V.B. *Performance Analysis of Carrier Sense Multiple Access with Collision Detection.* Computer Networks, Vol 4, No. 5, Oct/Nov 1980.

[12] Kleinrock, L. and Tobagi F.A. *Packet switching in radio channels: Part I, II, III, and IV.* IEEE Transactions on Communications, Vol. Com-25, Num 10, October 1977 (for part IV).

[13] Capetanakis J. I. *Tree algoritms for packet broadcast channels,* IEEE Transactions on Information Theory, vol IT-25, pp. 505-515, September 1979.

[14] Tsybakov B. and Mikhailov V. *Free synchronous packet access in abroadcast channel with feedback,* Prob. Inform. Trans., vol 14, pp. 259-280, Oct.-Dec. 1978.

[15] Fayolle G., Flajolet P., Hofri M. and Jacquet P. *Analysis of a Stack Algorithm for Random Multiple-Access Communication.* IEEE Transactions on Information Theory, Vol it-31, No 2, March 1985.

[16] Prucnal Paul R. and Santoro Mario A. *Spread Spectrum Fiber Optic Local Area Network.* Columbia University, Center for Telecommunication Research Technical Report.

[17] International Organization for Standardization. *Protocol for providing the connectionless-mode transport service,* ISO/DIS 8602, 1985.