Noname manuscript No.
(will be inserted by the editor)

RELD, RTT ECN Loss Differentiation to optimize the performance of transport protocols on wireless networks

Wassim Ramadan · Eugen Dedu · Dominique Dhoutaut · Julien Bourgeois

Received: date / Accepted: date

Abstract One major yet unsolved problem in wired-cum-wireless networks is the classification of losses, which might result from wireless temporary interferences or from network congestion. The transport protocol response to losses should be different for these two cases. If the transmission uses existing protocols like TCP, the losses are always classified as congestion losses by sender, causing reduced throughput. In wired networks, ECN (Explicit Congestion Notification) can be used to control the congestion through active queue management such as RED (Random Early Detection). It can also be used to solve the transport protocol misreaction over wireless networks. This paper proposes a loss differentiation method (RELD), based on ECN signaling and RTT (Round Trip Time), and applied to TCPlike. TCPlike is one of the three current congestion controls present in the new transport protocol DCCP (Datagram Congestion Control Protocol). Our simulations, using a more realistic simulated loss error model for wireless networks, show that RELD optimizes congestion control and therefore increases the performance of transport protocols over wireless networks, leading to an average performance gain ranging from 10% to 15%.

Keywords Wireless network · Transport Protocol · Congestion control · ECN · RED · RTT · Wireless packet loss model

1 Introduction

Wireless networks are now widely deployed and are commonly used to access services on the Internet in spite of lower performance noticed when compared to wired networks [2, 3]. Losses in wired networks are mainly due to congestion in routers, because congestion is usually handled by dropping the received packets when the router waiting queues are full or nearly full. Hence, losses in wired networks can be seen as an indication of congestion. This is different in wireless networks where losses often occur for various reasons, for example due to interference or poor link quality (high distance between the base station and the mobile device).

IEEE 802.11 already includes mechanisms to combat losses at the MAC layer. Wireless devices retransmit lost packets on a wireless link a certain number of times (6 for example). However, in case of long interferences, a packet can be lost 7 times consecutively on a wireless link. In this case, the device drops it and the transport level of the source discovers the loss. We are interested on loss processing at transport level.

The performance degradation reported on wireless networks appears because TCP (Transport Control Protocol) [24], commonly used by Internet applications and initially designed for wired networks, classifies any data loss as a congestion loss; therefore it reacts by reducing the transmission rate. However, in wireless networks,

W. Ramadan (contacting author)
Tel.: +33 3 8199 4787
Fax: +33 3 8199 4791
E-mail: wassim.ramadan@pu-pm.univ-fcomte.fr
· E. Dedu
Tel.: +33 3 8199 4775
E-mail: eugen.dedu@pu-pm.univ-fcomte.fr
· D. Dhoutaut
Tel.: +33 3 8199 4778
E-mail: dominique.dhoutaut@pu-pm.univ-fcomte.fr
· J. Bourgeois
Tel.: +33 3 8199 4775
E-mail: julien.bourgeois@pu-pm.univ-fcomte.fr
Laboratoire d'Informatique de l'Université de Franche-Comté (LIFC)
1 cours Leprince-Ringuet, 25200 Montbéliard, France

losses are not necessarily caused by congestion. There are many proposals on how to optimize the transport protocols performance on wireless networks in the literature; the main idea is that *transport protocols should reduce their transmission rate only in case of congestion and not if data is lost for other reasons* [3, 7, 5, 1].

Nowadays, more and more applications used over Internet, for example real-time media like audio and video streaming, can cope with a certain level of losses. If they use TCP, the high reliability may come at the price of great latency. UDP (User Datagram Protocol) [23], which does not have these drawbacks, lacks congestion avoidance support and flow control mechanisms. RTP (Real-time Transport Protocol) is an application protocol [28] widely used for streaming multimedia content (usually on the top of UDP). It allows the receiver to reorder received packets thanks to the sequence number included in the RTP packet header. RTP also uses a timestamp field which is useful in the context of real time applications synchronization. On the other hand, RTP, like UDP, does not deal with network conditions because it also lacks a congestion control.

Another promising protocol for these applications is DCCP (Datagram Congestion Control Protocol), recently standardized as RFC4340 [18], since it does not provide reliability but allows the use of congestion control protocols. One interesting point of DCCP is the freedom of choice for congestion control protocol: TCPlike [14], which reproduces the AIMD window progression of TCP SACK, or TFRC (TCP-Friendly Rate Control) [15]. As described in [18], DCCP implements bidirectional and unicast connections of congestion-controlled unreliable datagrams, and also:

1. negotiation of a suitable congestion control mechanism,
2. acknowledgement mechanisms for communicating packet loss and ECN (Explicit Congestion Notification) information, see section 5.1,
3. optional mechanisms that indicate to the sending application, with high reliability, which data packets reached the receiver, and whether those packets were corrupted, dropped in the receive buffer or ECN marked.

On the other hand, DCCP suffers from the same problem as TCP in wireless networks, meaning that any data loss is considered to be caused by congestion.

Because of all reasons mentioned before and because wired and wireless are often conjointly used, there is an increasing need for a new protocol that takes into account the properties of wireless links and the various reasons for data loss. In this paper we propose a new approach (*RELD, RTT ECN Loss Differentiation*) based on TCPlike over DCCP. It uses ECN in conjunction with RTT as the main factor to differentiate congestion losses from wireless losses, see section 5.3. RELD is an evolution of our previous method EcnLD [25] for loss differentiation, with an enhanced scheme that allows better realistic measurements than those obtained in classical ns2 simulator.

Contrary to some articles presented in related work (section 2), and also of our previous article [25], which used simple (homogeneous) error loss models for wireless links, the new results of this article show that in a real wireless environment where wireless losses are not uniform (see section 3 for the wireless loss model we use in this article), *RTT increases for wireless losses, and not for congestion losses*. In fact, an interference on the wireless channel will prevent the communication to continue throughout its duration. Hence, all packets sent during the interference time are buffered in the wireless access point. For each of them, a fixed number of MAC retransmissions is done, then the packet is dropped if it is still not acknowledged at MAC level. This buffering leads to an increasing of the RTT value for the packet which arrives just after the end of the interference. The results shown in 4.4 confirm this idea.

This paper is organized as follows. Section 2 presents related works on methods used to distinguish congestion losses from wireless losses. Section 3 presents the simulation environment, especially the wireless loss error model used. Section 4 shows the impact of congestion and wireless losses on the RTT, which helps to find out a formula for differentiating them. Section 5 presents RELD as a new method for loss classification. In section 6, performance of RELD is evaluated through extensive simulations. The article ends with the conclusions and some perspectives.

2 Related work

Many approaches have been proposed in the literature to differentiate losses. They are classified into three categories.

First category Certain approaches impose implementation of an intermediate agent between the source and the destination which is localized normally at the base station. Snoop [2, 3] is a TCP-aware link layer approach for local retransmission. It resides on a router or a base station and records a copy of every forwarded packet. Then, it inspects the ACK packets and carries out local retransmissions when a packet is corrupted by wireless channel errors. Other similar approaches, like ELN (Explicit Loss Notification) [1], can also be used to inform the sender that a loss has happened over wireless or wired networks. Although this kind of approach has a specific application field, it is necessary to make changes to the current base stations. Additionally, it needs more processing power at the base stations to process each packet.

Second category In this case, these approaches use end-to-end mechanisms. They do not require any network infrastructure changes. These methods can generally be classified into two main categories: those which depend on IAT (Inter Arrival Time) and those which depend on ROTT (Relative One-way Trip Time).

Parsa and Garcia in [22] consider losses as an indication of congestion if ROTT is increasing, and wireless losses otherwise.

Biaz [5] and its modified version **mBiaz** [9] use packets inter arrival time (IAT) at the receiver to classify losses. Biaz considers that when a packet arrives earlier than expected then a congestion loss has happened before. For wireless losses, the next packet arrives at around the time it should have, i.e. for n lost packets, if $(n+1)\,T_{min} \leq T_i < (n+2)\,T_{min}$ then the n packets are congestion losses. Otherwise, wireless losses.

mBiaz corrects an important misclassification for congestion losses. It makes a little modification to the high threshold of Biaz which becomes as follows:
$(n+1)\,T_{min} \leq T_i < (n+1.25)\,T_{min}$.

SPLD (Statistical Packet Loss Discrimination) [21] depends also on IAT. This scheme has a packet monitoring module to collect information about arriving packets. If during a certain time there are no losses, a statistical module updates the minimum IAT and the average. Then when losses occur, a discriminator module use IAT_{avg} to classify losses. SPLD considers that a loss is due to congestion if current IAT is greater than or equal to IAT stable (IAT_{avg}), otherwise it is a wireless loss.

Spike, derived from [30], is a method based on ROTT. In Spike, the packet is either in Spike state or not. A loss is considered a congestion loss in Spike state, and wireless loss otherwise. A packet enters Spike state when $ROTT > B_{spikestart}$, where $B_{spikestart}$ is the threshold indicating the maximum ROTT, and it leaves it if $ROTT > B_{spikeend}$, where $B_{spikeend}$ is the threshold indicating the minimum ROTT.

ZigZag [9], in addition to the deviation and the average of ROTT, is based on the number of losses n. If:

1. $n = 1$ and $rott_i < rott_{mean} - rott_{dev}/2)$, or
2. $n = 2$ and $rott_i < rott_{mean}$, or
3. $n = 3$ and $rott_i < rott_{mean} - rott_{dev}$, or
4. $n > 3$ and $rott_i < rott_{mean} - rott_{dev}/2$

then the n losses are considered as wireless losses, and congestion otherwise.

ZBS, described in [9], is a hybrid algorithm using ZigZag, mBiaz and Spike which chooses one of them depending on the following network conditions:
if $(rott < (rott_{min} + 0.05 * T_{min}))$ use Spike;
else if $(Tnarr < 0.875)$ use ZigZag;
else if $(Tnarr < 1.5)$ use mBiaz;
else if $(Tnarr < 2.0)$ use ZigZag;
else use Spike
where $Tnarr = T_{avg}/T_{min}$ (the average and the minimum inter arrival time).

TD (Trend and Loss Density based) [10] uses the trend of the ROTT and the density of losses. Authors observe that first, congestion losses often occur around and after a peak of ROTT curve and the network congestion last for a period of time after that. Second, rare are the cases when a congestion loss happens alone. Generally, a single packet lost is regarded as a wireless loss. So, TD uses loss trend to indicate if the packet loss happens around the ROTT peak curve or not and loss density to precise how often the loss occurs.

Finally, **Barma and Matta** in [4] is another end to end algorithm but uses the variance of RTT. Contrary to our results, they notice that RTT is high for congestion losses and low for wireless losses. In our opinion their results are based on a wrong assumption in the theoretical model[1] and in the simulation model used: To simulate a wireless network, a wired link with transmission errors was used[2], however the MAC retransmissions are not simulated, which means that the RTT increasing due to MAC retransmissions is not taken into account (see section 4 for detailed information).

Performance evaluation in [8] shows that methods based on ROTT perform better than those based on IAT because losses often appear around the peak of ROTT. Methods like Biaz and mBiaz have problems when several streams share the wireless link. Spike performs better than TD under the situation of low traffic but TD is better in case of high network congestion.

Third category Sender uses ECN (Explicit Congestion Notification) marking. Normal utilization of ECN to distinguish a congestion from a wireless loss works by testing the last interval of time in which a loss happened. If the source had previously received an ECN, then it indicates congestion, if not, it indicates a wireless loss. TCP-Eaglet [6] authors consider that ECN

[1] "The basic tenet of our approach is that if the packets are suffering congestion losses, the observed RTTs will vary but if packets are suffering random losses, the observed RTTs will not vary much".

[2] 'These [wired] links represent wireless links with transmission errors".

marking does not work all the time for classification losses. They propose to halve sending rate when either TCP is in Slow Start phase and there is one or more losses, or TCP sender has an ECN indication in Congestion Avoidance phase as a response to imminent congestion.

Another method, similar to TCP-Eaglet, is ECN-D [31]. According to ECN-D, two scenarios are possible:

1. there are only wireless losses in the current congestion window (cwnd)
2. wireless losses occur simultaneously with congestion losses

For the first scenario, ECN-D finds out that a wireless loss occurred because of the non presence of ECN notification. So a loss is considered as congestion loss if and only if there is an ECN mark. Additionally, for better performance, the value of cwnd at the sender is reduced only once per window in presence of ECN marks. ECN-D is proposed to optimize the SCTP performance which does not support the use of ECN messages.

Our results show that TCP-Eaglet and ECN-D are not efficient differentiation schemes because they do not take into account congestion losses without ECN mark. However, as our RELD belongs to the same category, we evaluate in this article the performance of RELD with regard to TCP-Eaglet (same idea as ECN-D).

3 Simulation environment

Simulations tools (such as NS2) are convenient when evaluating and tuning new protocols. However, one should be particularly careful when choosing their many parameters when doing simulations. In this work, it is mandatory to dispose of realistic lower levels models - radio propagation and medium access control, as we are interested in the impact of the real radio environment on DCCP communications and the ways to overcome the resulting problems.

3.1 Propagation and loss models

From the point of view of a higher level protocol such as DCCP, only lost packets are really taken into account (thanks to the layering of network protocols, DCCP is not designed to be aware of radio attenuation, collisions or interferences). However, the way these losses appear, their frequency and their distribution over time have a great impact on the behavior of DCCP and the enhancements we are proposing in this paper. Different ways of simulating realistic losses exist. A first - simple - one would consist in using a traditional radio propagation model (such as the well known *tworayground* or *shadowing* models of NS2), and then add a loss or error model which drops a certain amount of packets. This is a perfectly viable solution, but we have chosen instead to use a more realistic propagation model, which will cause all the losses by itself, as it can be much more easily linked or derived from a real environment.

3.1.1 Existing radio propagation models

Many models do exist, each one having its own strengths and drawbacks. Depending of the context, one has to be particularly careful when choosing the model used. The following list presents the main families of models that can be found in the literature :

– Models using a continuous attenuation equation of the radio signal and taking only the distance into account. The model proposed by Friis [16] handles propagation in completely obstacle-free environments, and other models propose faster attenuation depending on distance, such as the *tworayground* which considers a two-path propagation causing self alteration to the received signal.
– Models derived from the preceding ones, integrating a kind of *fast-fading* causing packet drops. Some use the Gilbert-Elliot model (a Markov chain determining whether a packet should be dropped or not depending on the current state) causing a very fast variation of the link quality. Another derived model called *shadowing* proposes a random factor in the quality of each packet on top of the attenuation caused by the distance. There is a probability of losing a packet, and this probability grows with the distance. Setting parameters for such a model can be done using empirical data obtained from real measurements (for example, the attenuation factor will be high in indoor environment, medium in a city and low in an open field, and the standard deviation can also be obtained from experiments). In [32], they defined a multi-path model combined with a Doppler effect in order to calculate the Bit Error Rate (BER) in function of the signal-noise ratio and speed.
– More complex models do exist, which require modeling the whole environment (or at least the most relevant parts of it) with its obstacles. *Raytracing* techniques are used to calculate the signal level depending on the relative positions of transmitter and receiver in this complex environment. The involved algorithms are computationally-intensive but can be pre-calculated for a given environment [29].

– Even more advanced propagation algorithms are available, not only using numerous obstacles, but also their composition or surface properties which greatly affect radio propagation. With those models not only are occlusions caused by obstacles modeled, but so are refraction and diffraction, enabling detailed propagation calculations for indoor environments [27].
– Lastly, some models take a different approach and essentially use real collected data. [13]. Their main advantage is to propose very realistic values, but which are of course strongly tied to a particular environment.

3.1.2 Shadowing-pattern

The simpler models (*Friss, tworayground, shadowing, ...*) are not detailed nor realistic enough for us to use here, in particular because of their simplistic losses distributions over time. On the other and, models which require an extremely detailed modeling of the environment (for use with raytracing or similar techniques) produce results that are only valid in the specific context that was modeled.

Because of these concerns, in this paper we decide to use the *shadowing-pattern* model described in [12] and [11], which is based on the standard *shadowing* but add the ability to change the signal strength in a bursty way, which in turn produces statistically realistic bursty losses. The *shadowing-pattern* model was originally designed for use in vehicular ad hoc networks, but is quite versatile.

This model mimics the reality where different elements in the environment have cumulative effects on the strength of the signal at the receiver. It makes use of so-called *perturbators*, which are an abstraction of real-world elements (or group of) that have an impact on the signal strength. Those elements can be as varied as peoples or cars passing by, doors opening and closing, other nodes sending data over the radio channel, etc. *Perturbators* affect a limited and configurable area of the virtual environment. They alternate between two states, *active* and *inactive*.

– In *active* state, they affect - usually reducing the strength - any signal received by a node inside the area of effect of the *perturbator*.
– In *inactive* state, they have no effect at all

Each perturbator is thus defined by the time and the standard deviation of the time spent in both states, along with its strength when active. Figure 1 shows the state-graph of two *perturbators* . Their effect value (in dBm) when in active state is simply a modifier which

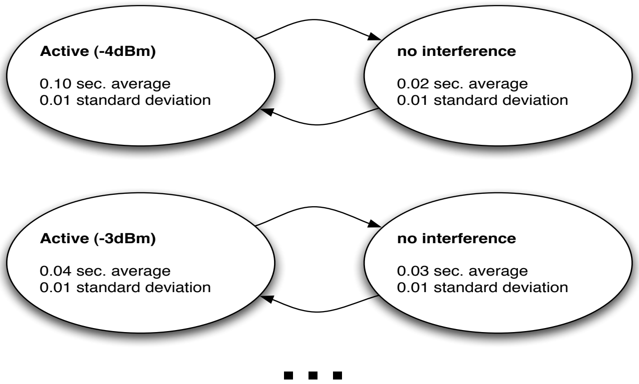

Fig. 1 State-graph of two perturbators

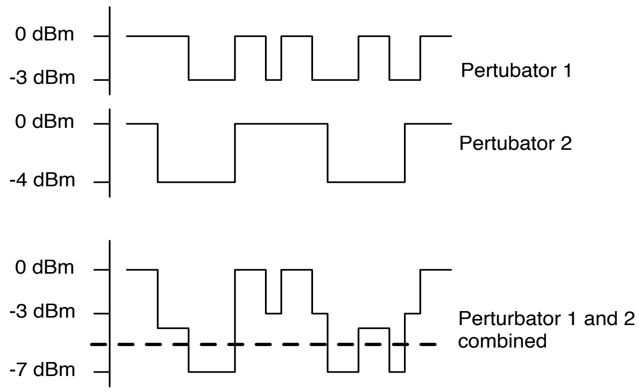

Fig. 2 Effects of perturbators over timer and combination

is added to they signal at the receiver - which already takes distance into account, being itself computed by a Friis-like equation. Figure 2 shows how their individual effects evolve over time and also how they can be combined and sometimes pass a threshold they would not have been able to if taken individually.

As explained in [12], this techniques does not aim for an accurate modeling of a particular environment. It instead aims at producing extremely realistic statistical behaviors, particularly in term of losses distribution over time. It is also easy to configure and requires lightweight computations.

In fact, when using *shadowing-pattern*, one has just to choose a set of perturbator and configure their parameters. As their effects combine, a set of a few (2 to 5) perturbators is usually enough to describe a quite realistic environment. And as any average and standard deviation can be given, it can be used to model real world phenomena of any time scale.

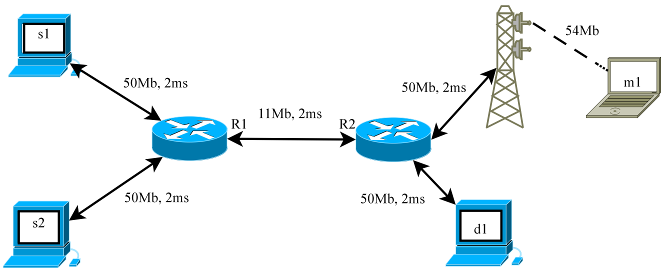

Fig. 3 ns2, network topology.

3.2 Simulation topology

Figure 3 shows the dumbbell topology employed for carrying out the simulations. It is a wired-cum-wireless topology with 2 senders (s1 and s2) and 2 receivers (m1 and d1). The link between routers R1 and R2 is the bottleneck for wired network. The wireless network uses the standard 802.11 for wireless communications, with a bandwidth of 54Mbit/s. The detailed parameters can be found in appendix A. Each node (routers, access point and edge nodes) uses RED for queue management with default values. ECN is enabled on all of them. The packet size is 500 bytes and the simulation time is 60 seconds.

3.3 Configuring the physical network

The simulator is ns2 [20] version 2.34. In the next sections, in order to evaluate our propositions in a range of realistic contexts, we will run 51 different tests over this topology, changing the parameters of the radio propagation model.

The difference between the 51 tests is the level of wireless perturbation added to the wireless network. Perturbations on the wireless channel are performed using *shadowing-pattern* perturbators. A mix number of zero to three out of seven perturbators is used in each of these tests. Table 1 shows these seven perturbators, their power, when they are active and when they are not. As the simulations presented in this paper are focused on standard WiFi networks (a mobile computer accessing the network through a base-station), and as we are interested in DCCP flows, we will only use relatively high frequency *perturbators*, with individual effects lasting no more than one hundredth second. It is obvious that phenomena that could prevent any communications in the network for many seconds or even hours are beyond the scope of DCCP flow control optimization. Also keep in mind that even no individual *perturbator* effect lasts for more than a few hundredths of a second, multiple perturbator effects and durations can overlap. In such (common) event, they effectively prevent communications for a longer period.

Because of the chosen fixed topology, all perturbators taken alone except number 7 had no effect on the reception threshold of wireless channel. When two of them are combined and active the signal attenuation brings the wireless signal under the reception threshold which translates to packet loss on wireless channel. Perturbators number 1 and 2 have the same power but with different time effect (2 is stronger than 1). Same thing for 3 and 4 (4 is stronger than 3). So, in total we have one test without perturbation and 50 tests with a number of one to three mix perturbators. This variety of tests aims at producing a group of realistic wireless environments, ranging from absolutely not to very disturbed radio channels.

4 Influence of losses on RTT

Losses are traditionally caused by buffer filling on routers. However, nowadays wireless networks are ubiquitous, and in these networks losses are numerous and increase with signal degradation according to various environment circumstances: distance between mobile and base station, high bit error rate related to wireless link and so on. Such losses induce sender to halve its sending rate, because it wrongly considers losses as a sign of congestion, which is not always the case. For better performance the sender should be able to distinguish congestion from wireless losses. The purpose of this section is to show that the RTT can be used to differentiate losses.

But before this, the next section analyses the usefulness of loss differentiation.

Note that the RTT used throughout this article is the *RTT of the packet following the lost packet* (time between ack reception and corresponding data packet sending), which gives information about the lost packet.

4.1 Mathematical study on usefulness of loss differentiation

In this section a mathematical study is presented to analyse the effect of loss classification on the overall throughput. The model shown here is derived from [17, 4] and adapted to AIMD (additive increase multiplicative decrease) algorithm of TCPlike. In TCPlike congestion control, the ACK ratio (denoted here by R), corresponding to the frequency of ACKs for received packets, is defined as a parameter[3].

[3] In DCCP, R can change during a communication. In TCP, R is fixed and equals 1 or 2.

perturbator number	*power* (dBm)	*inactive time* (sec.)	*standard deviation* (sec.)	*active time* (sec.)	*standard deviation* (sec.)
1	-2	0.03	0.005	0.01	0.01
2	-2	0.01	0.02	0.03	0.01
3	-3	0.03	0.005	0.01	0.01
4	-3	0.01	0.02	0.03	0.01
5	-4	0.03	0.01	0.02	0.01
6	-5	0.04	0.01	0.02	0.01
7	-6	0.04	0.01	0.01	0.01

Table 1 The seven perturbators.

On each received ACK, the congestion window (cwnd) is increased by $R/cwnd$ when $cwnd \geq ssthresh$. This means that the congestion window is increased by one packet for every window of data acknowledged without lost or marked packets. On the other hand if the ACK reports lost or marked packets, cwnd is divided by 2 ($cwnd = cwnd/2$).

Suppose that p is the packet drop probability for which cwnd is halved (which usually is the drop probability on overall wired and wireless networks), and RTT is the round trip time. Then the expected change of cwnd on each received ACK will be:

$$E[\Delta cwnd] = \frac{(1-p)*R}{cwnd} - \frac{cwnd*p}{2} \qquad (1)$$

In case of one ACK each R received packets, the time between each two updates of cwnd is $\frac{R*RTT}{cwnd}$. So, the rate change $x(t)$ in this laps of time is:

$$\frac{dx(t)}{dt} = \frac{\left(\frac{(1-p)*R}{cwnd} - \frac{cwnd*p}{2}\right)RTT}{\frac{R*RTT}{cwnd}} \qquad (2)$$

This differential equation can be written like this:

$$\frac{dx(t)}{dt} = \frac{1-p}{RTT^2} - \frac{p}{2R}x^2(t) \qquad (3)$$

Let $a = \frac{1-p}{RTT^2}$ and $b = \frac{p}{2R}$. Then by integration we have:

$$\int_0^{x(t)} \frac{1}{a-bx^2(t)}\,dx(t) = \int_0^t dt \qquad (4)$$

which gives:

$$\frac{\tanh^{-1}\left(\sqrt{\frac{b}{a}}x(t)\right)}{\sqrt{ab}} = t + c \qquad (5)$$

$$\frac{\ln\left(1+\sqrt{\frac{b}{a}}x(t)\right) - \ln\left(1-\sqrt{\frac{b}{a}}x(t)\right)}{2\sqrt{ab}} = t + c \qquad (6)$$

$$x(t) = \sqrt{\frac{a}{b}} * \frac{e^{2t\sqrt{ab}+C}-1}{e^{2t\sqrt{ab}+C}+1} \qquad (7)$$

The steady state throughput of TCPlike is given by:

$$x = \lim_{t\to\infty} x(t) = \sqrt{\frac{a}{b}} \qquad (8)$$

By replacing a and b with their values we obtain the influence of p on the throughput:

$$x = \frac{1}{RTT}\sqrt{\frac{2R(1-p)}{p}} = \frac{1}{RTT}\sqrt{2R\left(\frac{1}{p}-1\right)} \qquad (9)$$

We now compare equation 9 in case of loss classification and without classification. Let p_c be the probability that a packet is dropped because of congestion, and p_w the probability that a packet is dropped on the wireless link. In the case where wireless losses do not halve the congestion window cwnd (loss differentiation is used), $p = p_c$. In the classical case, no differentiation is used, so $p = p_c + p_w$. Let's denote also the throughput of the classical method by x_c and the throughput of the loss differentiation method by x_l. Then, the expected gain of the throughput is:

$$\frac{x_l}{x_c} = \frac{\frac{1}{RTT}\sqrt{2R\left(\frac{1}{p_c}-1\right)}}{\frac{1}{RTT}\sqrt{2R\left(\frac{1}{p_c+p_w}-1\right)}} = \sqrt{\frac{\frac{1}{p_c}-1}{\frac{1}{p_c+p_w}-1}} \qquad (10)$$

From equation 10 we can conclude that:

1. if $p_w = 0$ then $x_l = x_c$
2. if $p_w > 0$ then $x_l > x_c$ and the ratio increases while p_w increases.

Otherwise said, the loss differentiation usefulness increases while the number of wireless losses p_w increases.

4.2 The impact of loss type on the RTT in theory

Impact of a congestion loss on RTT: Let s be the time needed for a router to process and send a packet, i.e. the service time of the queue. Suppose a packet is enqueued in a router queue (see figure 4). Let n be the place in the queue in the case the previous packet has been enqueued. If, on the contrary, the previous packet has been dropped, then the packet is placed at position $n-1$, hence it takes s less time to be processed by the router. Otherwise said, after a packet drop, this router reduces the RTT of the packet by a time equal to s.

An example of value for s is given in the following. Suppose a router with a 100Mb/s interface, and

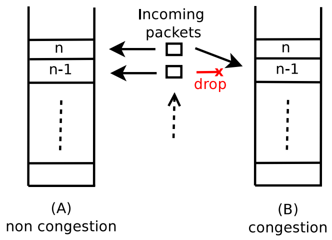

Fig. 4 Theoretical impact of congestion losses on RTT.

1000 bytes packets. The interface sends at 100Mb/s = 12.8MB/s = 12.8kpkt/s. This means that a packet takes 1/12.8ms, so $s \approx 0.1$ms. For higher speed interfaces, s is smaller. Figure 5 shows similar results, with differences of less than 1ms generally.

To conclude, the RTT of a packet *decreases* after a congestion loss.

Impact of a wireless loss on RTT: When a loss occurs in a wireless network, it is retransmitted at MAC level until either it is received, or the retry limit is reached. If retry limit is reached, the packet is simply dropped.

In order to reduce wireless network collision, for each MAC retransmission the wireless card waits, according to the standard, a certain number of slots (called a *backoff*). A slot s equals $20\mu s$, and the number is taken randomly in the interval between 0 and the value of contention window (CW). CW is initialized to $2^5 - 1$ for the first attempt. If the card does not receive an ACK for the sent packet, CW doubles (without however exceeding 1023) and the transmission is tried again. This is done for each retransmission. So for the first retransmission CW=$2^6 - 1$, for the second retransmission CW=$2^7 - 1$, and generally for the n-th retransmission, CW=$\max(2^{5+n} - 1, 1023)$.

In a real world environment, and especially if the receiver is close to the range limit, the channel conditions can be bad enough to prevent multiple successive transmission attempts. If all MAC retransmissions fail, the packet is lost. Consequently, for a packet lost on the wireless network[4], the backoff contributes with an additional time of $s \cdot \sum_{i=1}^{i=n} \max(rand(2^{5+i-1} - 1), 1023)$. This time is added to the RTT of the next packet to arrive, which has waited in the queue. In case of a retry limit equal to 7 (this value is used in real networks and also in ns2 network simulator, and means 1 transmission and 6 retransmission at maximum) the additional time is equal in average to:
$20\mu s \times (31 + 63 + 127 + 255 + 511 + 1023 + 1023)/2 =$ 30.33ms.

[4] Compared to the smallest transmission time (a packet which succeeded at the first transmission and with backoff equal to 0).

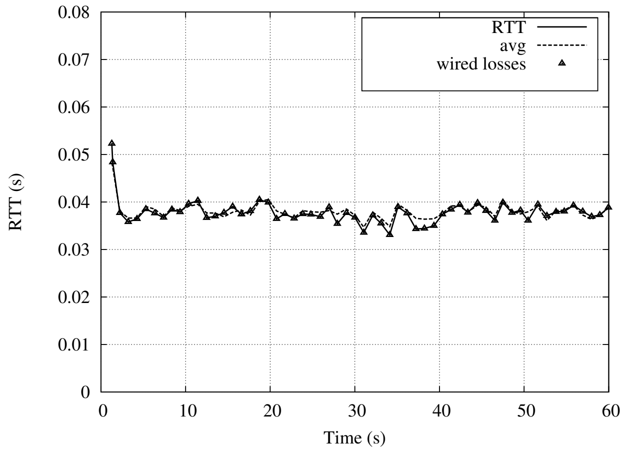

Fig. 5 Impact of congestion losses on RTT.

As such, the RTT increases by about 30ms for a wireless loss[5]. Figure 6 shows similar results, for example at second 30 the difference between average RTT and the RTT after a wireless loss is 15ms and at second 32 the same difference is 35ms.

To conclude, the RTT of a packet *increases* after a wireless loss.

4.3 The impact of loss type on the RTT in simulation

To evaluate the impact of loss type (congestion or wireless) on the RTT in simulation, we use the network presented in section 3.2, and the *shadowing-pattern* propagation model presented in section 3.1.2. We present in this section the result of two tests, with a strong perturbator and without any perturbator, both of them using the original TCPlike congestion control under DCCP in ns2.

Figure 5 presents the RTT evolution in presence of congestion losses; in this figure no perturbator is used, hence no wireless losses are present. It can be noticed that the RTT is generally stable (generally between 0.034 and 0.04 seconds). Also, most of congestion losses have an RTT smaller than the average, as can be seen at 37, 38 and 39 seconds for example.

Figure 6 presents the RTT evolution in presence of congestion losses and wireless losses; in this figure a perturbator (number 7) for wireless channel is used. It can be seen that the wireless perturbation makes the RTT unstable (generally between 0.025 to 0.07 seconds). Second, most of congestion losses appear when RTT is below average i.e. between 53 and 57 seconds, while most

[5] This is the backoff contribution only; other factors, less important in our study, can modify this value, such as the transmission itself or other transmissions during backoff waiting.

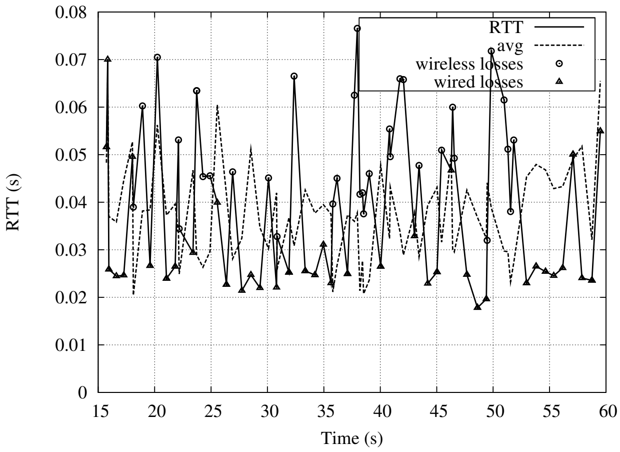

Fig. 6 Impact of congestion and wireless losses on RTT.

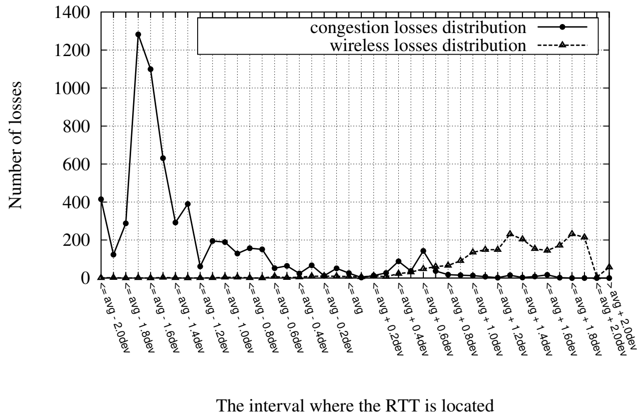

Fig. 7 Distribution of losses based on RTT intervals.

of wireless losses appear when RTT is above average i.e. at 20, 22 and 30 seconds.

The previous two sections, the theory and the simulation results, lead to the same conclusion:

- RTT for congestion losses are generally *smaller* than average RTT
- RTT for wireless losses are generally *greater* than average RTT

Moreover, it appears that the difference of RTT is usually greater for wireless losses than for congestion losses.

4.4 The choice of RTT threshold to distinguish losses

The previous sections showed that it is possible to classify losses using their RTT values. This section further analyses the RTT evolution for congestion and wireless losses with respect to average RTT and its deviation. It aims to find out a threshold between congestion and wireless losses.

The results are presented as an average of the results of all the 51 tests depicted previously. Figure 7 presents the distribution of congestion and wireless losses around the average RTT (avg) using a step of one tenth of the RTT deviation (dev). First, this figure confirms that congestion losses have generally RTTs smaller than avg, and wireless losses have generally RTTs higher than avg. Moreover, most of congestion losses appear between $avg - 1.8dev$ and $avg - 1.3dev$, and wireless losses appear between $avg + dev$ and $avg + 1.9dev$.

The RTT threshold is better viewed using cumulative RTTs. In figure 8, each bar indicates the percent of congestion losses which have an RTT smaller than the value on the x axis. It can be noticed that for the 51 tests about 90% of the congestion losses have an

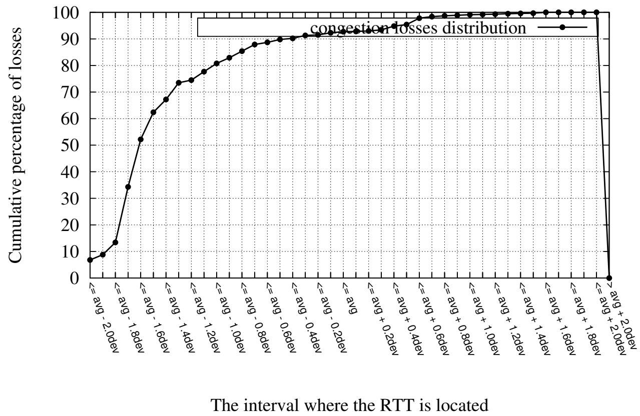

Fig. 8 Cumulative distribution of congestion losses.

RTT $\leq avg$. It is about 98% at $avg + 0.6dev$, and all of them have an RTT $\leq avg + 1.5dev$.

Figure 9 presents the same distribution, but for wireless losses. It can be seen that 3% of wireless losses have an RTT $\leq avg$. The percentage is 8% for RTT $\leq avg + 0.6dev$ and more than 60% for RTT $\leq avg + 1.5dev$.

The two figures show also that it is much more frequent to have congestion losses with RTT $\geq avg$ than wireless losses with RTT $\leq avg$.

These figures show that perfectly classifying all losses using RTT is impossible. However, choosing a threshold between avg and $avg + 0.6dev$ can correctly classify the majority of congestion losses and misclassify only a few wireless losses (between 3% to 8%).

We have tested a threshold of avg and of $avg + 0.6dev$. The results were similar, albeit a bit worse for avg. As the value of 0.6 is chosen somewhat empirically, we cannot sustain that it is the *best* value for all the cases. Nevertheless, we sustain that the best threshold

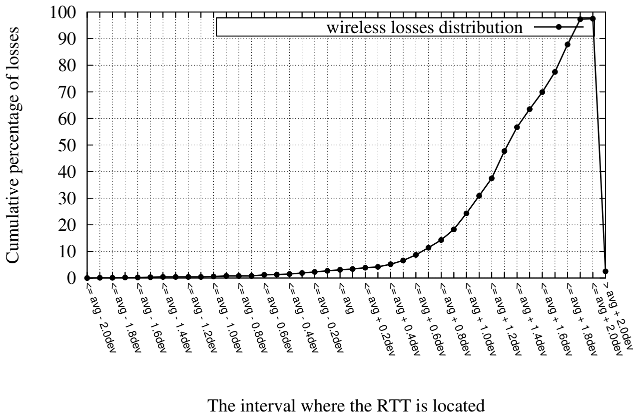

Fig. 9 Cumulative distribution of wireless losses.

should be a bit greater than avg. In the remaining of the article we will use a threshold of $avg + 0.6dev$.

5 RELD, RTT ECN Loss Differentiation

The purpose of RELD is to use ECN in conjunction with RTT to prevent network congestion and to maintain sending rate in case of wireless losses. To differentiate between congestion losses and wireless channel losses, RELD requires that intermediary routers between sender and receiver are ECN compatible. For this, it is necessary that the routers implement an active queue management such as RED (Random Early Detection).

5.1 ECN principle

ECN is an extension of IP (Internet Protocol) defined in RFC3168 [26] which works with active queue managements such as RED (described in the next section) and which supports an end-to-end congestion notification without losing packets. It is optional and it is only used when both connection endpoints want to use it. In this case, an ECN compatible router updates a field in the IP header of packets to indicate imminent congestion. When the receiver finds out that a packet was marked, it indicates this ECN information to the sender in its acknowledgement. The sender reacts to ECN signal as if the packet had been lost.

5.2 Active queue management, RED

Nowadays, an active queue management such as RED (Random Early Detection) is implemented in many routers. Using RED leads to better sharing among the various flows passing through the router. RED is also used for congestion management through negative feedback to the sender, which is done by dropping packets before queue overflows in order to signal imminent congestion. If utilization of ECN is enabled in the router and flow is ECN capable, RED marks these packets instead of dropping them. To do it, RED maintains a few values: queue length q_l, queue average q_{ave}, minimum queue threshold q_{th_min} and maximum queue threshold q_{th_max}.

– If $q_{ave} < q_{th\_min}$, all packets pass without being dropped or marked.
– If q_{ave} is between q_{th_min} and q_{th_max}, packets are marked with a probability which increases while q_{ave} increases.
– Finally, when $q_{ave} > q_{th\_max}$ all packets are dropped.

5.3 RELD details

Like our method, TCP-Eaglet [6] uses ECN information. However, it does not deal with losses in Slow Start phase (the first phase of a TCP connection, where the congestion window increases exponentially with the RTT) and hence it does not consider the case where a burst of packets arrive suddenly to a router and exceed its queue capacity. In this case, there may be a significant number of ECN unmarked losses, which might appear even in Congestion Avoidance phase (the long-running phase of a TCP connection, where the congestion window increases linearly with the RTT) if other concurrent flows are in Slow Start phase.

The contribution of this paper is that, unlike TCP-Eaglet, RELD takes these situations into account. First, it makes no difference between Slow Start and Congestion Avoidance phase. Then, it uses the RTT to remedy the ECN weakness, as shown below.

As ECN marking occurs often before congestion, a responsive sender to ECN can use this information to prevent congestion and to differentiate congestion losses from wireless losses. A sender which reduces its sending rate in response to ECN can avoid congestion in most cases but not all. In fact, when a burst of packets arrives to the router, its queue might become full. Since the queue average has not changed much, the router drops packets without marking them. In such cases, losses are numerous and they often causes an RTT growth.

To sum up, we consider that a loss is due to congestion if and only if:

1. $ecn > 0$
   or
2. $n > 0$ and RTT $< avg + 0.6dev$

where *ecn* is the number of packets marked EC (Experienced Congestion), n the number of lost packets indicated by the received Ack, RTT the current RTT, avg the average RTT and dev the RTT deviation.

In this manner, RELD works as TCPlike in increasing phases, i.e. in Slow start and in Congestion Avoidance phases the congestion window will increase as usually. On the other side, when the sender receives a loss indication it will decrease its congestion window only if the losses are considered by RELD as a congestion (as described on the above formula). Formula 2 has been deduced from the results of the previous section. The complexity of formula 2 is constant, like the other loss differentiation algorithms in related work, because the calculations of RTT, avg and dev are made by a simple equation using constant values and just two saved values each time.

6 Simulation results

This section, through extensive simulations, analyses RELD threshold and its classification rate, shows the performance gain of RELD compared to original TCPlike, and compares RELD to TCP-Eaglet. For this, two scenarios are used: without competition and with competition of another flow.

We made a small modification to ns-2 so that RED and ECN can be used on wireless links. For DCCP protocol in ns2, we used the patch written by Mattson [19] and currently maintained by us[6].

6.1 Verification of the RTT threshold chosen by RELD

In section 4.4 an RTT threshold has been chosen from statistical results so that it allow to distinguish efficiently between congestion and wireless losses. The congestion control protocol used for that was TCPLike. The goal of the current section is to validate this choice by using RELD as congestion control protocol under DCCP and verifying that this new algorithm does not alter the classification.

It should be noted that in the following figures, differentiation is done by a thorough parsing of the log files. An indeed strong point of using a simulator is that it can precisely tell us where and why packets where really lost.

6.1.1 Scenario without competition

Results of this analysis are shown in figures 10, 11 and 12. Figure 10 confirms the two conclusions from section 4: congestion losses have an RTT smaller than average RTT, while wireless losses are generally above average RTT.

[6] `http://eugen.dedu.free.fr/ns2`

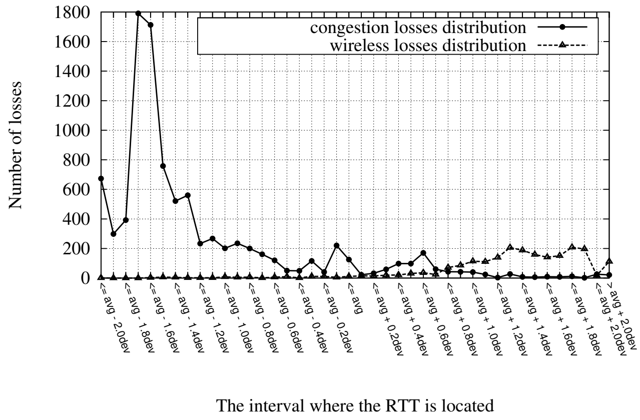

Fig. 10 RELD, without competition: distribution of losses based on RTT intervals.

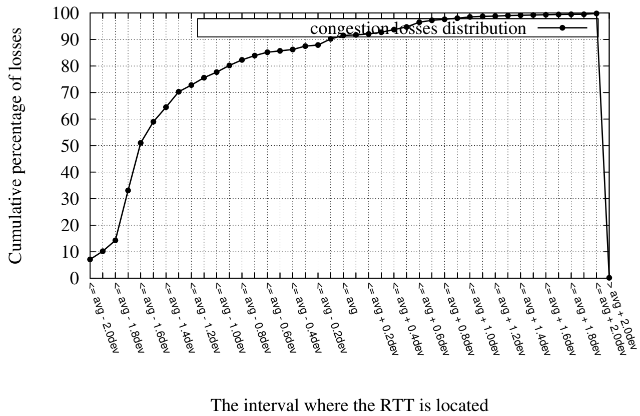

Fig. 11 RELD, without competition: cumulative distribution of congestion losses.

Figures 11 and 12 consolidate the choice of $avg + 0.6dev$ in case where there is no competition with other flows. As shown in figure 11, only 2% of congestion losses are not included in RELD formula (have RTT $\geq avg + 0.6dev$). For wireless losses, figure 12 shows that about 10% of them are not included in RELD formula.

6.1.2 Scenario with competition

A concurrent TCP flow is added to the network (figure 3), between s2 as a sender and d1 as a receiver, and it appears twice: from 1 to 20 seconds and from 25 seconds to 45. Its goal is to create traffic in Slow Start mode (when the queues are likely to become full,

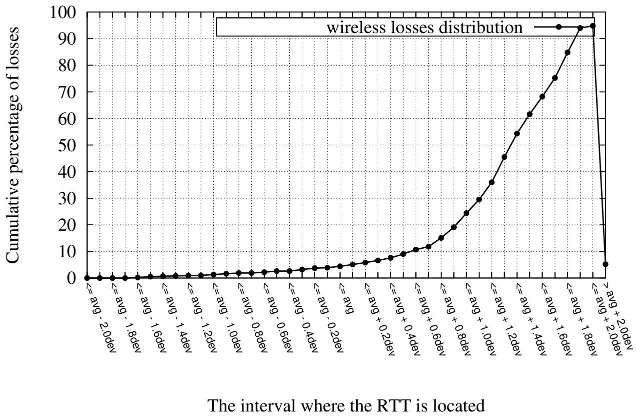

Fig. 12 RELD, without competition: cumulative distribution of wireless losses.

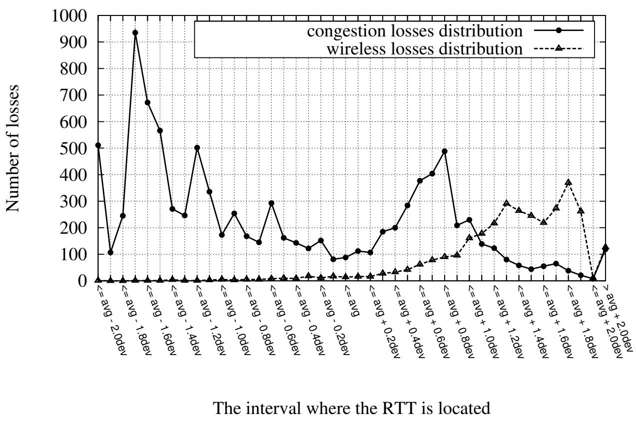

Fig. 13 RELD, in competition: distribution of losses based on RTT intervals.

and without ECN notification) several times during the simulation.

Only the results of the 51 tests for the RELD flows are presented in the three figures. Figure 13 presents the distribution of losses. First, it shows that fewer congestion losses are gathered to the left margin of the graphic; in fact, congestion losses between $avg - 1.8dev$ and $avg - 1.6dev$ in figure 13 are twice fewer than those of the same interval in figure 10 (900 compared to 1800). Second, congestion losses are more evenly distributed. Third, congestion losses span more to the right, which means that the loss classification is more difficult. And fourth, as before, most of congestion losses are at the left and wireless losses are at the right.

Also, figure 14 shows that fewer congestion losses (70%) are included in the RELD formula (RTT $\leq avg + 0.6dev$). Of course, choosing a greater threshold can reduce congestion losses misclassification, however this will result in a higher wireless misclassification rate.

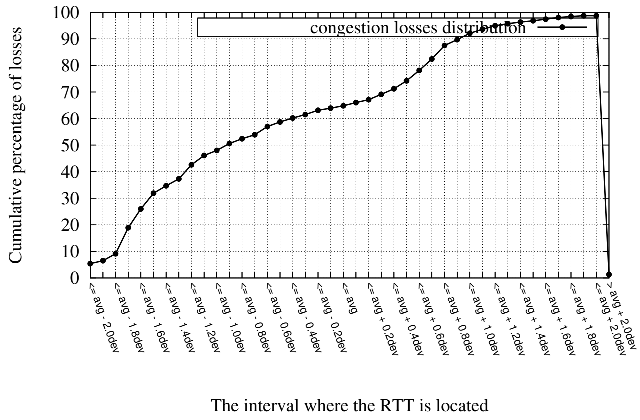

Fig. 14 RELD, in competition: cumulative distribution of congestion losses.

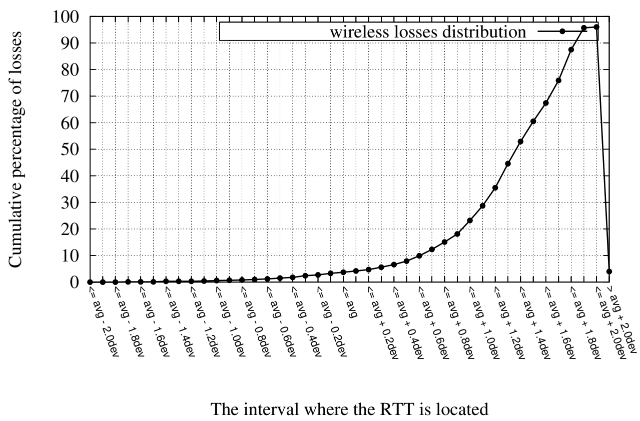

Fig. 15 RELD, in competition: cumulative distribution of wireless losses.

Finally, for wireless losses presented on figure 15, the same RTT distribution is noticed, which is normal because the same perturbators are used.

The conclusion of this section is that RELD threshold allows to classify congestion losses correctly between 80% (in case of competition with other traffic) and 98% (in case without competition).

6.2 Evaluation of RELD classification accuracy

In this section we evaluate RELD performance through its ability to classify congestion and wireless losses using the formula which combine ECN and RTT. This performance is evaluated for a wide rage of channel conditions; from absolutely not to very disturbed. Those conditions are expressed by the identifiers of the perturbators used for a given simulation. "0" means only perturbator 0 was used, "234" means perturbators 2, 3 and 4 were used in conjunction, etc.

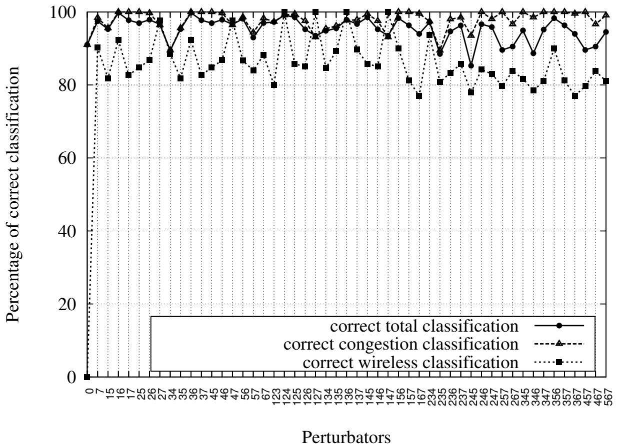

Fig. 16 RELD, without competition: correct classification rate.

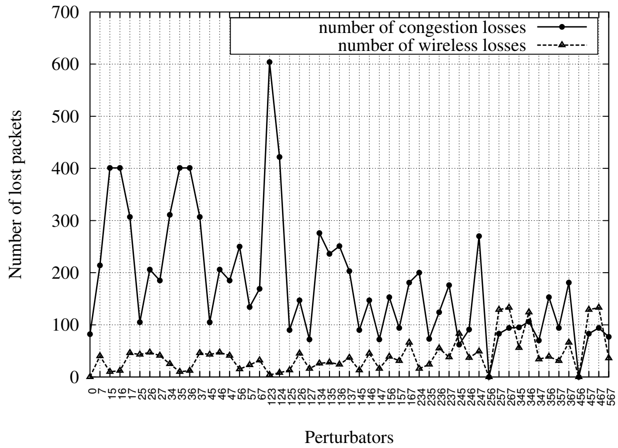

Fig. 17 RELD, without competition: number of congestion and wireless losses.

The classification makes use of several parameters, explained in the following. Suppose that during a test the following results are obtained: there were 80 congestion losses and 20 wireless losses. Among the congestion losses, 70 of them are correctly identified as congestion losses, and 10 wrongly identified as wireless losses. For wireless losses, 5 are wrongly classified and 15 correctly classified. This is resumed in the following table (c means congestion losses, w means wireless losses):

real	80c	20w
classified	70c 10w	5c 15w

The percentage of correct congestion loss classification is 70/80, and of correct wireless loss classification is 15/20. The percentage of total correct classification is the total number of correctly classified losses divided by the total number of losses: $(70 + 15)/(80 + 20)$.

In the following figures, the real reason for each packet loss (congestion or wireless loss) is taken from the simulation trace file, and the considered reason for loss is printed by the source of the flow, which uses RELD classification, from the ns2 source code.

6.2.1 Scenario without competition

Figure 16 shows three curves: first one present percentage of correct classification for total lost packets including congestion and wireless losses, second and third shows correct classification for congestion and wireless separately. The total correct classification varies between 85% and about 99% in most cases, and it is about 92% in average. Correct congestion classification in this scenario without competition is very high thanks to RELD threshold which covers majority of congestion losses. On the other hand, correct wireless classification rate, while smaller than congestion one, is high too, varying between 78% and 100%.

Figure 17 gives an idea about how many lost packets arrived in each simulation test. As shown in this figure, the number of congestion losses is very small (up to 600 losses, compared to between 15000 and 115000 sent packets, depending on the test), which means that the ratio of received to sent packets is very high. In other words this high ratio is very important for applications with special needs, such as multimedia streaming, since it avoids the need of packet retransmission methods; moreover, a high number of lost packets leads to quality degradation.

6.2.2 Scenario with competition

As previously, a TCP concurrent flow is added in the network, between s2 and d1. Contrary to figure 16, figure 18 shows that correct classification rate of wireless losses is higher than similar rate for congestion losses. As shown in 6.1.2, this result is normal due to RELD threshold which tries to balance between congestion and wireless perturbation. In all tests the rate of correct classification is very high, around 80% in average for total and for congestion losses. The correct rate increases when the wireless losses number increases, in tests where perturbators numbers are higher than 123.

The number of congestion and wireless lost packets is given in figure 19. Compared to figure 17, it shows two things: first, the ratio of received packets is very high too; second, the number of congestion losses is smaller, due to the bandwidth sharing with another traffic.

6.3 RELD vs TCPlike

A loss classification method is supposed to make transport protocol perform better in wireless environment.

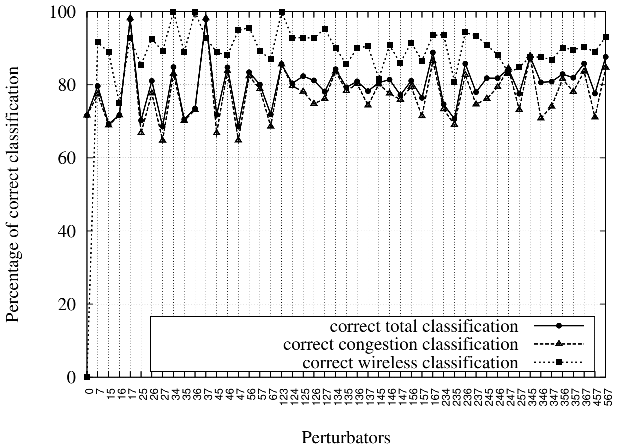

Fig. 18 RELD, in competition: correct classification rate.

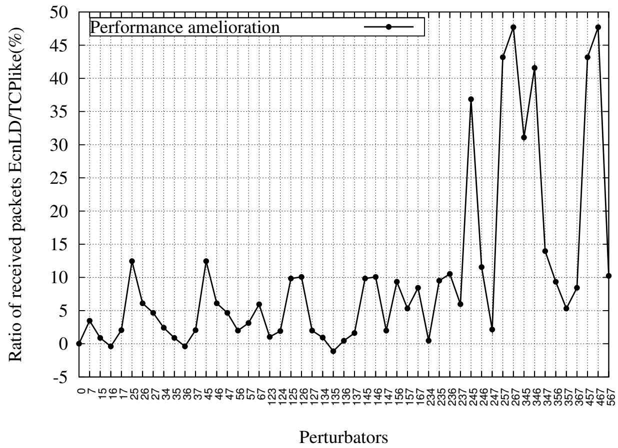

Fig. 20 RELD vs TCPlike, without competition: performance amelioration.

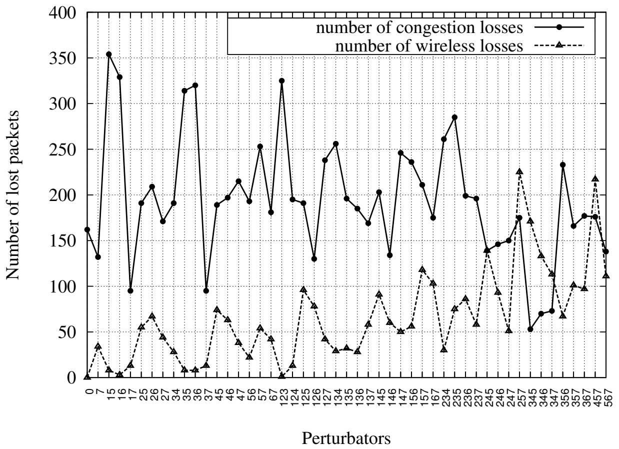

Fig. 19 RELD, in competition: number of congestion and wireless losses.

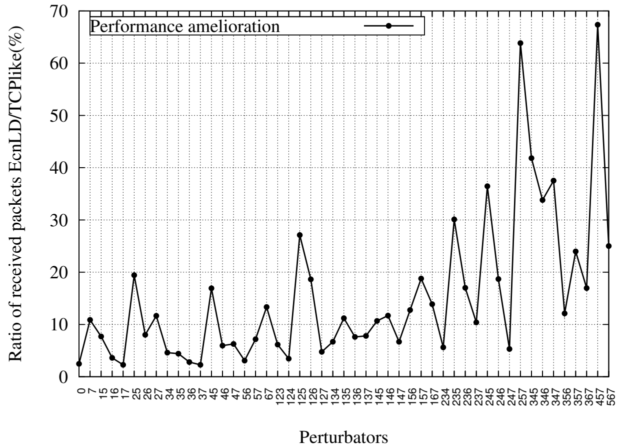

Fig. 21 RELD vs TCPlike, in competition: performance amelioration.

To verify the performance amelioration brought to DCCP via RELD loss classification algorithm we compare the number of received packets by receiver when using TCPlike and RELD. Results are expressed in ratio between the number of received packets by the receiver of RELD and the same number for receiver of TCPlike. The ratio value indicates the amelioration, equality or the degradation of TCPlike performance.

6.3.1 Scenario without competition

Comparison results of the 51 tests are shown in figure 20. First, it shows that, except a very few cases, the ratio received/sent packets is greater for RELD than for TCPlike. Second, the amelioration is high, with an average of 10% and a maximum of 47%. Third, the performance gain is higher when the number of wireless losses is higher, i.e. perturbators sets 257, 267, 346, 457 and 467 have the highest number of wireless losses (as shown previously in figure 17) and the highest performance gain. On the contrary, the lowest amelioration appears when the number of congestion losses is high and the number of wireless losses is low, i.e. perturbators sets 35, 36 and 135.

6.3.2 Scenario with competition

As previously, a TCP concurrent flow is added in the network, between s2 and d1. Figure 21 shows the results. The same conclusions as in the scenario without competition apply here, with a higher average amelioration of 15% and maximum of 68%.

The conclusion is that RELD ameliorates the performance of transport protocol both without and with competition. Performance gain is higher when wireless losses are higher, and is smaller when congestion losses are higher.

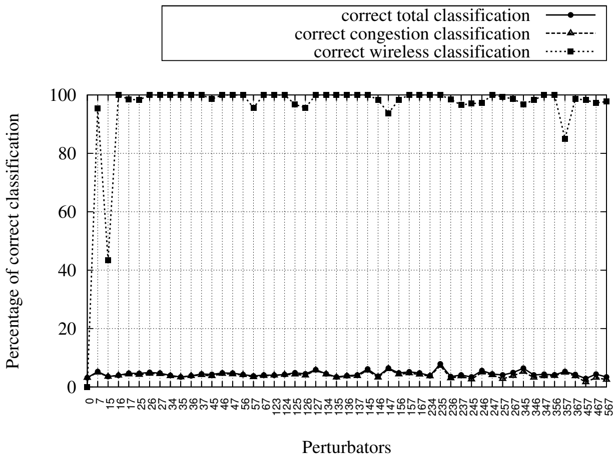

Fig. 22 Eaglet, without competition: correct classification rate.

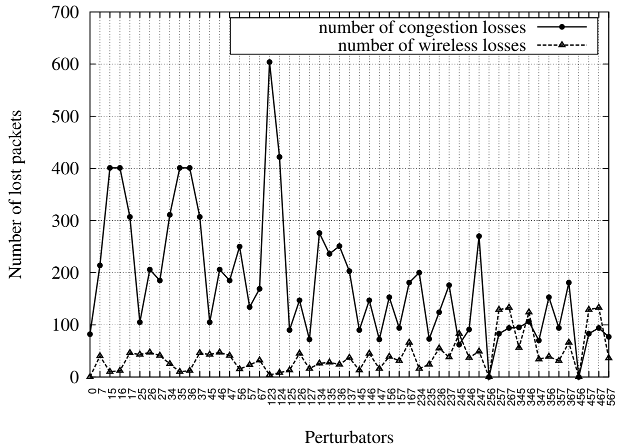

Fig. 23 Eaglet, without competition: number of congestion and wireless losses.

6.4 RELD vs TCP-Eaglet

As mentioned in related works (section 2), third category, there is also TCP-Eaglet which uses ECN to distinguish between congestion losses and wireless ones. It is based on a very strong assumption: since ECN is able to prevent congestion in wired networks, it is also able to distinguish losses. For it, a lost packet in Congestion Avoidance mode means wireless loss, while an ECN marked packet means congestion. Our results show that this hypothesis is often not valid, especially in a perturbed wireless environment, and as such can lead to congestion in the network.

Original TCP-Eaglet was designed for TCP, not for DCCP. However, we are interested in comparing our approach with its idea, so we implemented it for DCCP in ns2. In this section, we compare RELD and TCP-Eaglet in a scenario without competition. As for RELD (figures 16 and 17), the correct classification results and number of congestion and wireless losses are presented (figures 22 and 23). Because of the high number of lost packets, as shown in figure 23, where between 4000 and 18000 packets are lost, correct classification rate is very small, about 4.4% in average. On the other hand, TCP-Eaglet has an average of 95% of wireless loss classification, which is not a satisfactory result knowing that correct classification of congestion losses is on the contrary very low (most of losses are classified as wireless).

As the results of TCP-Eaglet are already not good, we do not test its performance in the scenario with competition.

Discussion For all tests, the default parameters of RED were used. TCP-Eaglet indicates that these parameters must be adjusted to fit their assumption, which could be one of the reasons of the bad results of TCP-Eaglet.

A second reason is that, in wired networks, an ECN marked packet has a negligible chance to be lost afterwards. On the contrary, this can arrive in a wireless network, known to have sometimes a high number of losses. Figure 24 shows such an example, where sender s1 sends 3 packets, p_x1, p_x2 and p_x3 and router R1 marks p_x2 as congestion experienced but they are all dropped on the wireless channel because of an interference. When a previously ECN marked packet is lost on a wireless link, the ECN notification does not arrive at the sender, which will continue to increase its sending rate as usually. As shown in section 5.2, RED has three states (four states for gentle RED): all packets pass, packets are marked probabilistically, packets are dropped. If by bad luck all the packets *marked* during $q_{th\_min} < q_{ave} < q_{th\_max}$ period are lost by the wireless channel afterwards, then RED enters $q_{ave} > q_{th\_max}$ period, when all the following packets are dropped (without marking/notification). For these latter packets, since the TCP-Eaglet sender has not received any ECN marked packet before, it continues to increase the sending rate as usually, leading to many losses.

On the other hand, RELD copes with this problem thanks to the other indicator (the RTT). In this case, when marked packets are lost on a wireless channel, the RTT will increase, and RELD acts like TCP-Eaglet. However, for the latter losses, as they are congestion losses, their RTT decreases, and RELD considers them as congestion losses, hence it decreases its sending rate.

This shows that using ECN alone to distinguish losses is not satisfactory. Moreover, RELD sender reacts to congestion faster than TCP-Eaglet.

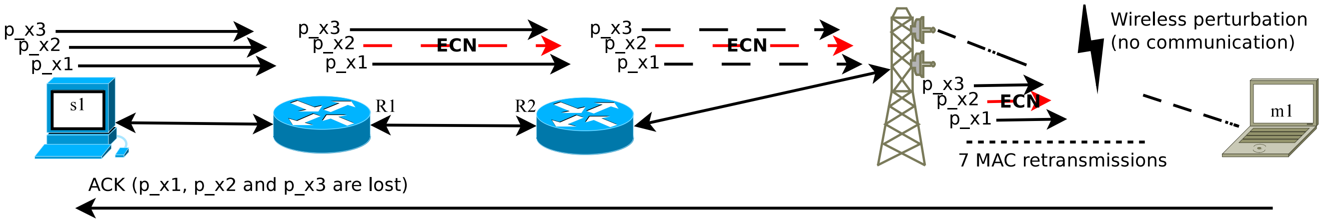

Fig. 24 A marked packet can be lost on a wireless channel afterwards.

7 Conclusion

In this article, we have proposed a general method to solve some issues related to low performance of transport protocols in wireless networks. This study has shown that congestion control is more efficient on wireless networks if the loss classification is correctly made between losses due to wireless media and losses due to congestion. We have also shown that ECN can successfully be used with RTT to differentiate congestion losses from wireless losses. Moreover, our statistical results done by simulations, which use a more realistic loss error model for wireless networks, confirm that RTT increases in case of wireless losses. This confirmation is contrary to some studies about loss differentiation. We recommend the use of RELD for video streaming over wireless networks because the reception rate obtained by RELD is very high (the majority of packets are received). We still have tracks to be followed in this particular study, first, to analyze our method in more different environment conditions and more competition traffic. Second, to monitor the effect of ns2 propagation models on the results of loss differentiation methods. And finally, to evaluate the performance of RELD for transmission of real video.

Acknowledgements Wassim Ramadan has a grant of PhD thesis from the Ministry of High Education of Syria.

References

1. Balakrishnan H, Katz R (1998) Explicit Loss Notification and Wireless Web Performance. In: IEEE GLOBECOM Global Internet
2. Balakrishnan H, Seshan S, Katz RH (1995) Improving reliable transport and handoff performance in cellular wireless networks. Wireless Networks 1(4):469–481
3. Balakrishnan H, Padmanabhan VN, Seshan S, Katz RH (1997) A comparison of mechanisms for improving TCP performance over wireless links. IEEE/ACM Transactions on Networking 5(6):756–769
4. Barman D, Matta I (2002) Effectiveness of loss labeling in improving TCP performance in wired/wireless networks. In: ICNP, IEEE Computer Society, pp 2–11
5. Biaz S, Vaidya NH (1999) Discriminating congestion losses from wireless losses using inter-arrival times at the receiver. IEEE Symposium ASSET pp 10–17
6. Biaz S, Wang X (2004) Can ECN be used to differentiate congestion losses from wireless losses? Tech. Rep. CSSE04-04, Auburn University
7. Biaz S, Wang X (2004) RED for improving TCP over wireless networks. In: ICWN: International Conference on Wireless Networks, pp 628–636
8. Boukerche A, Jia G, Pazzi RWN (2007) Performance evaluation of packet loss differentiation algorithms for wireless networks. In: PM2HW2N: Performance monitoring and measurement of heterogeneous wireless and wired networks, ACM, pp 50–52
9. Cen S, Cosman PC, Voelker GM (2003) End-to-end differentiation of congestion and wireless losses. IEEE/ACM Transactions on Networking 11(5):703–717
10. Chou CF, Hsu MW, Lin CJ (2006) A trend-loss-density-based differential scheme in wired-cum-wireless networks. In: IWCMC: international conference on Wireless communications and mobile computing, pp 239–244
11. Dhoutaut D, Spies F (2007) Adding geographical interferences into the shadowing pattern model for vehicular ad hoc networks simulations. In: ITST '07. 7th International Conference on Intelligent Transport Systems Communications
12. Dhoutaut D, Régis A, Spies F (2006) Impact of radio propagation models in vehicular ad hoc networks simulations. In: VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks, ACM, New York, NY, USA, pp 40–49, DOI http://doi.acm.org/10.1145/1161064.1161072
13. Eude N, Ducourthial B, Shawsky M (2005) Enhancing ns-2 simulator for high mobility ad hoc networks in car-to-car communication context. In: Proceedings of the 7th IFIP International Conference on Mobile and Wireless Communications Net-

works (MWCN 2005), Morocco
14. Floyd S, Kohler E (2006) Profile for Datagram Congestion Control Protocol (DCCP) congestion control ID 2: TCP-like congestion control. RFC 4341
15. Floyd S, Kohler E, Padhye J (2006) Profile for Datagram Congestion Control Protocol (DCCP) congestion control ID 3: TCP-friendly rate control (TFRC). RFC 4342
16. Friis H (1946) A note on a simple transmission formula. Proc IRE 34
17. Kelly F (2001) Mathematical modelling of the Internet. In: Mathematics Unlimited — 2001 and Beyond, Springer-Verlag, Berlin, Germany, pp 685–702
18. Kohler E, Handley M, Floyd S (2006) Datagram Congestion Control Protocol (DCCP). RFC 4340
19. Mattsson NE (2004) A DCCP module for ns-2. Master's thesis, Luleå University of Technology
20. ns (2009) The Network Simulator NS-2 (v2.34). http://www.isi.edu/nsnam/ns/
21. Park MK, Sihn KH, Jeong JH (2006) A statistical method of packet loss type discriminartion in wired-wireless network. In: CCNC: IEEE Consumer Communications & Networking Conference
22. Parsa C, Garcia-Luna-Aceves JJ (1999) Improving TCP congestion control over internets with heterogeneous transmission media. In: ICNP: IEEE International Conference on Network Protocols, pp 213–221
23. Postel J (1980) User Datagram Protocol. RFC 768
24. Postel J (1981) Transmission control protocol. RFC 793
25. Ramadan W, Dedu E, Bourgeois J (2009) EcnLD, ECN loss differentiation to optimize the performance of transport protocols on wireless networks. In: International Conference on Ultra Modern Telecommunications & Workshops (ICUMT), WMCNT workshop, IEEE, Saint Petersburg, Russia, 1, pp 1–6
26. Ramakrishnan K, Floyd S, Black D (2001) The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168
27. De La Roche G, Rebeyrortte R, Jaffrès Runser K, Gorce JM (2005) A New Strategy for Indoor Propagation Fast Computation with MR-FDPF Algorithm. In: Acta Press (ed) Antennas, Radar and Wave Propagation, IASTED, Banff Canada, URL http://hal.inria.fr/inria-00405192/en/
28. Schulzrinne H, Casner S, Frederick R, Jacobson V (2003) RTP: A Transport Protocol for Real-Time Applications. RFC 3550
29. Stepanov I, Herrscher D, Rothermel K (2005) On the impact of radio propagation models on manet simulation results. In: Proceedings of the 7th IFIP International Conference on Mobile and Wireless Communication Networks (MWCN 2005), Marrakech, Morocco
30. Tobe Y, Tamura Y, Molano A, Ghosh S, Tokuda H (2000) Achieving moderate fairness for UDP flows by path-status classification. In: LCN: IEEE Local Computer Networks, p 252
31. Ye G, Saadawi T, Lee MJ (2004) Improving stream control transmission protocol performance over lossy links. Selected Areas in Communications, IEEE Journal on 22(4):727–736
32. Yin J, ElBatt T, Yeung G, Ryu B, Habermas S, Krishnan H, Talty T (2004) Performance evaluation of safety applications over dsrc vehicular ad hoc networks. In: in Proceedings of VANET, pp 1–9

A Simulation parameters

The Ns2.34 parameters used for the simulations were the following:

Parameter name	Parameter value
Mobile distance	20m
Mac/802_11: basicRate_	54Mbs
Mac/802_11: PLCPDataRate_	54Mbs
Mac/802_11: dataRate_	54Mbs
Mac/802_11: RTSThreshold_	3000
Queue/RED: setbit_	true
node-config: -ifqLen	Queue/RED
node-config: -adhocRouting	DSDV
node-config: -propType	TSShadowingPattern
TSShadowingPattern: pathlossExp_	4
TSShadowingPattern: std_db_	0

Biography of the authors

Wassim Ramadan is PhD student at the Computer Science laboratory of the University of Franche-Comté, France. His thesis aims to improve the performance of transport protocols in wireless networks to make them more adapted to video transfer. He has a special interest in video streaming in wireless networks.

Eugen Dedu is Assistant Professor at the Computer Science laboratory of the University of Franche-Comté at Montbéliard, France. He obtained his PhD in Computer Science at the University of Versailles Saint-Quentin-en-Yvelines in 2002. His main research interests are in the transport level in computer networks and MEMS sensor/actuators networks.

Dominique Dhoutaut is Assistant Professor at the Computer Science laboratory of the University of Franche-Comté, France. He obtained his PhD in Computer Science at INSA of Lyon in 2003. His general research interests are in radio and ad hoc networking, and in particular Vehicular Ad Hoc Networks (VANET).

Julien Bourgeois is Professor at the Computer Science laboratory of the University of Franche-Comté, where he leads the research team on complex networks. His interests include multimedia streaming, optimization of P2P and MEMS sensor/actuators networks and security management in complex networks.