

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

Wassim Ramadan, Eugen Dedu, Julien Bourgeois
Laboratoire d'Informatique de l'Université de Franche-Comté (LIFC)
{Wassim.Ramadan, Eugen.Dedu, Julien.Bourgeois}@pu-pm.univ-fcomte.fr

Abstract—One major yet unsolved problem in wired-cum-wireless networks is the classification of losses, which can be due either to wireless temporary interferences or to network congestion. The transport protocol response to losses has to be different for these two cases. If the transmission uses existing protocols like TCP, the losses will always be classified as congestion losses by the data 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 resolve the transport protocol misreaction on wireless networks. This paper proposes a loss differentiation method (EcnLD), based on ECN signaling and RTT, and applied to TCPlike. TCPlike is one of the two current congestion controls present in the new transport protocol DCCP (Datagram Congestion Control Protocol). Our results indicate that EcnLD is a good approach to optimize congestion control and therefore increase the performance of transport protocols over wireless networks.

Index Terms—Wireless network, Transport Protocol, Congestion control, ECN, RED.

I. INTRODUCTION

Wireless networks are now widely deployed and are often used to access services on the Internet in spite of lower reported performance when compared to wired networks [1], [2]. 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, poor link quality, or the distance between the base station and the mobile device.

There are already mechanisms for loss processing at the link layer. Wireless devices retransmit lost packets on a wireless link a certain number of times (7 for example). However, a packet can be lost 7 times consecutively on a wireless link (an example is a long interference). In this case, the device gives up and the transport level of the source discovers the loss. We are interested on loss processing at transport level.

The reason for the performance degradation reported on wireless networks is because TCP (Transport Control Protocol) [3], commonly used by Internet applications and designed generally for wired networks, classifies any data loss as a congestion loss, and therefore it reduces the transmission rate. However, in wireless networks, losses are not necessary due

to 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* [1], [4], [5], [6].

Nowadays, more and more applications transported over Internet, for example real-time media like audio and video streaming, can accept a certain level of losses. If they use TCP, they have to pay the price of high reliability, such as sometimes great latency. UDP (User Datagram Protocol) [7], which does not have these drawbacks, lacks congestion avoidance support and flow control mechanisms. RTP (Real-time Transport Protocol) is an application protocol [8] widely used for streaming multimedia (usually on the top of UDP). It gives to the receiver the possibility to reorder received packets thanks to the sequence number included in the RTP packet header. RTP uses also a timestamp field which is very useful in real time application for synchronization purposes. On the other hand, RTP, like UDP, does not deal with network conditions because it lacks also a congestion control.

Another promising protocol for these applications is DCCP (Datagram Congestion Control Protocol), recently standardized as RFC4340 [9], 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 [10], which is similar to TCP SACK, or TFRC (TCP-Friendly Rate Control) [11].

As described in [9], 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 III-A,
- 3) optional mechanisms that tell 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 due to congestion.

Because of this and because wireless networks are widely deployed, 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 (**EcnLD, ECN loss differentiation**) based on TCPlike over DCCP. It uses ECN as the main factor to differentiate congestion losses from wireless losses. It also uses the RTT (Round Trip Time) to optimize the lack of ECN as a discriminator between wireless losses and congestion losses, see section III-C.

This paper is organized as follows: Section III presents EcnLD as a new method for loss classification. In section IV, performance of EcnLD is evaluated through simulations. Section II presents related works on methods used to distinguish congestion losses from wireless losses. Finally, section V concludes this article and presents some perspectives.

II. RELATED WORKS

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

a) First category: Here, some approaches impose implementation of an intermediate agent between the source and the destination which is localized normally at the base station. Snoop [2], [1] is a TCP-aware link layer approach for local retransmission. It resides on a router or base station and records a copy of every forwarded packet. Then, it looks into 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) [6], can also be used to inform sender that a loss has happened over wireless or wired networks. Even if this kind of approach is very interesting, 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.

b) Second category: Here, approaches use end-to-end mechanisms. They do not require any changes to the network infrastructure. These methods can be generally classified into two main categories: methods depending on IAT (Inter Arrival Time) and on ROTT (Relative One-way Trip Time):

Biaz [5] and its modified version **mBiaz** [12] 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 been arrived, 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) [13] depends also on IAT and 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 [14], is a method based on ROTT. In Spike, the packet is either in Spike state or not. If it is and there is a loss then it is a congestion loss, otherwise it is a wireless loss. 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 [12], 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 [12], 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) [15] 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 [16] is another end to end algorithm but uses the variance of RTT (Round-trip delay time). They notice that RTT is high for congestion losses and low for wireless losses.

Performance evaluation in [17] 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.

c) Third category: Sender uses ECN (Explicit Congestion Notification) marking, see next section. 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 has previously received an ECN, then it indicates congestion, if not, it is a wireless one. TCP-Eaglet [18] authors consider that ECN marking does not work all the time for classification losses. They propose to halve sending rate when either TCP is in Slow Start mode and there is one or more losses, or TCP sender has an ECN indication

in congestion avoidance mode as a response to imminent congestion. As our EcnLD is placed in the same category, we have evaluated performance of EcnLD with regard to TCP-Eaglet. Another proposition using ECN is given in [19]. A loss is considered as congestion loss if and only if there is an ECN mark. Additionally, for better performance, the sender cwnd is reduced only once per window in presence of ECN marks. However, this is not an efficient differentiation scheme because it does not take into account losses without ECN mark.

III. ECNLD, ECN LOSS DIFFERENTIATION

To differentiate between congestion losses and wireless channel losses, EcnLD 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).

A. ECN principle

ECN is an extension of IP (Internet Protocol) defined in RFC3168 [20] which works with 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 bit 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.

B. 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} .

- 1) If $q_{ave} < q_{th_min}$, all packets pass without being dropped or marked.
- 2) If q_{ave} is between q_{th_min} and q_{th_max} , packets are marked with a probability which increases while q_{ave} increases.
- 3) Finally, when $q_{ave} > q_{th_max}$ all packets are dropped.

C. EcnLD details

To our knowledge, TCP-Eaglet [18] is *the only method in the literature which uses ECN for loss classification*. TCP-Eaglet does not deal with losses in Slow Start mode and hence it does not consider the case where a burst of packets arrive suddenly to a router and exceeds its queue capacity. In this case, there may be a significant number of ECN unmarked losses, which might appear even in Congestion Avoidance mode if other concurrent flows are in Slow Start mode.

The contribution of this paper is that, unlike TCP-Eaglet, EcnLD takes these situations into account. First, it makes no difference between Slow Start and Congestion Avoidance mode. 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 $RTT_{cur} > RTT_{ave} + RTT_{dev}$

where ecn is the number of packets marked EC (Experienced Congestion), n the number of lost packets indicated by the received Ack, RTT_{cur} the current RTT, RTT_{ave} the RTT average and RTT_{dev} the RTT deviation.

In this manner, EcnLD works as TCPlike in increasing phases, i.e. in Slow start and in Congestion avoidance modes 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 EcnLD as a congestion (as described on the above formula). In fact, as shown in related works also, RTT increases in case of congestion. Formula 2 has been deduced empirically, by comparing RTT_{cur} to several values between $RTT_{ave} - RTT_{dev}$ and $RTT_{ave} + RTT_{dev}$. Best results have been obtained by formula 2. The complexity of formula 2 is constant because the calculations of RTT_{cur} , RTT_{ave} and RTT_{dev} are made by a simple equation using constant values and just two saved values each time.

IV. PERFORMANCE MEASUREMENTS

A. Simulation model

Figure 1 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 11Mbit/s for some tests or 54Mbit/s for others. 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 50 seconds.

Tests are done using the simulator ns2 version 2.31 [21]. We also made a small modification to ns-2 so that RED and ECN can be used on wireless links. We evaluate the performance of EcnLD compared to original TCPlike, measure the accuracy of EcnLD and compare the results of classification to TCP-Eaglet. To do this, we added an error model on the wireless

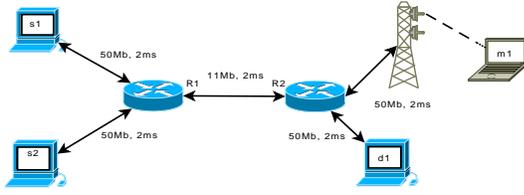


Fig. 1. ns2, network topology

network which varies from 0% to 10% (this model is applied on the incoming and the outgoing wireless link; thus a $N\%$ of loss rate in the error model gives about $2N\%$ of real wireless error rate). We also implemented TCP-Eaglet. All this on a version of DCCP written by Mattson [22] and maintained by us¹ in ns2. The next section present our simulation results.

B. Simulation results

1) *EcnLD vs TCPlike*: We compare the loss classification performance of EcnLD and TCPlike. Each simulation is repeated eleven times, once without error rate and ten times with a real error rate which varies from 2% to 20% on the wireless network. The same test is performed twice by changing the maximum number of MAC retransmissions (one and two retransmissions at the MAC layer in addition to the initial transmission) and for each type of congestion control, EcnLD and TCPlike. The purpose of this change is to obtain a significant number of wireless losses which happens in the reality but not in ns2. The sender of this simulation is the node s1 which creates a connection to the mobile m1 on the wireless side. We present the results in two cases: first, without having a competition on the network. Then, in presence of a TCP flow from s2 to d1. This latter flow 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) several times during the simulation.

a) *Scenario without competition*: In this scenario, the tested flow benefits of all available resources alone. Figures 2 and 3 show the results of comparison between EcnLD and TCPlike in terms of data packets sent and received by the EcnLD and TCPlike sender/receiver. It may be noted in Figure 2 that EcnLD takes over and that its superiority increases with the increase of loss number. Indeed, EcnLD receiver receives about twice more data packets than TCPlike receiver for an error rate of 20%. The difference between these two congestion control decreases with the increasing of number of retransmissions. This is due to the very small number of wireless losses because of the retransmission of erroneous packets by the access point. It should be noted that EcnLD has a better performance in the case of one MAC retransmission than in case of two. This is because the number of MAC retransmissions has a non-negligible influence on the RTT.

b) *Scenario in competition with a TCP flow*: Figures 4 and 5 show again that EcnLD has an advantage over TCPlike.

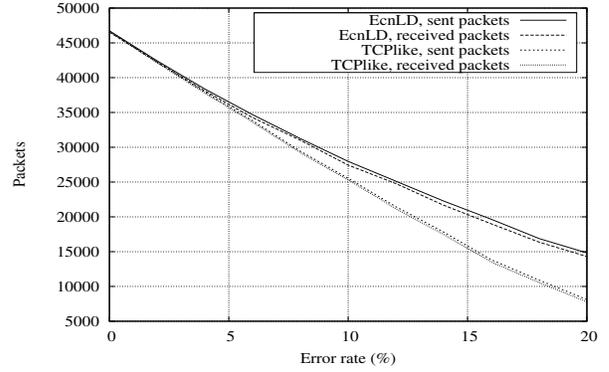


Fig. 2. EcnLD vs TCPlike for one MAC retransmission without concurrence.

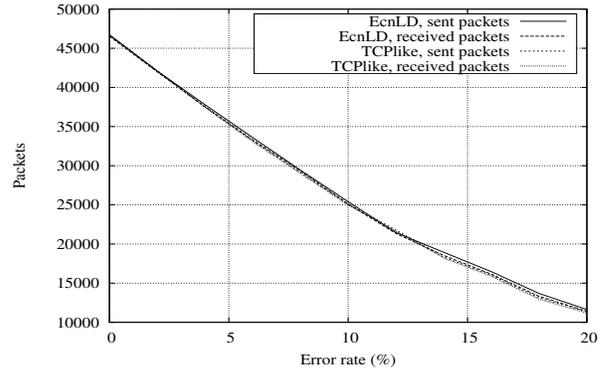


Fig. 3. EcnLD vs TCPlike for two MAC retransmissions without concurrence.

This advantage is quite significant for one retransmission in figure 4, where more packets are received by m1. EcnLD shows a greater performance when the wireless error rate is greater than 5%. This difference is much less important for two retransmissions due to the very small number of wireless losses in this case. When there are no errors on the wireless network (0%), the performance is the same as expected.

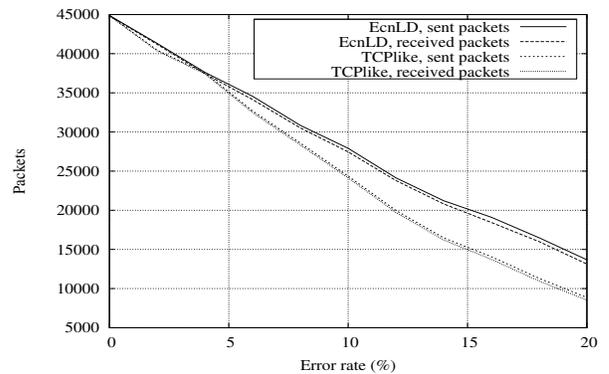


Fig. 4. EcnLD vs TCPlike for one MAC retransmission with concurrence.

2) *EcnLD vs Eaglet*: Knowing that TCP-Eaglet is not proposed to be used with DCCP, we are interested to validate our approach in relation to authors hypothesis. We compare

¹<http://lifc.univ-fcomte.fr/~dedu/ns2>

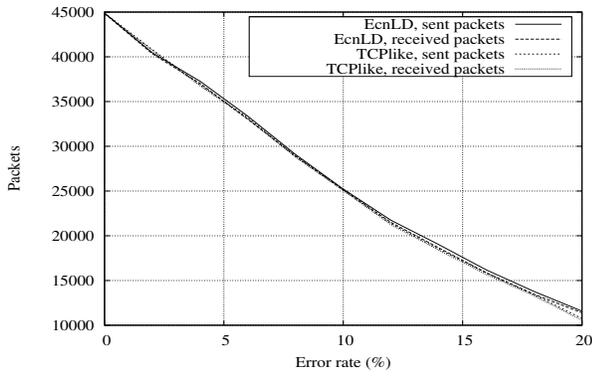


Fig. 5. EcnLD vs TCPlike for two MAC retransmissions with concurrence.

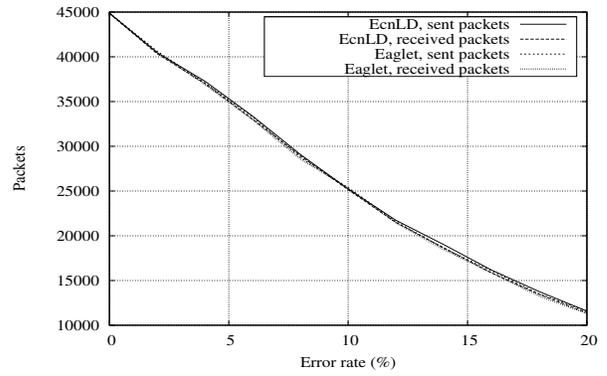


Fig. 7. EcnLD vs Eaglet with competition on a wireless network of 11Mb/s and two retransmissions MAC.

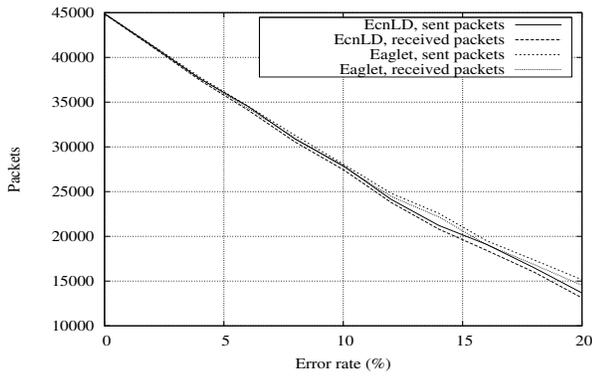


Fig. 6. EcnLD vs Eaglet with competition on a wireless network of 11Mb/s and one MAC retransmission.

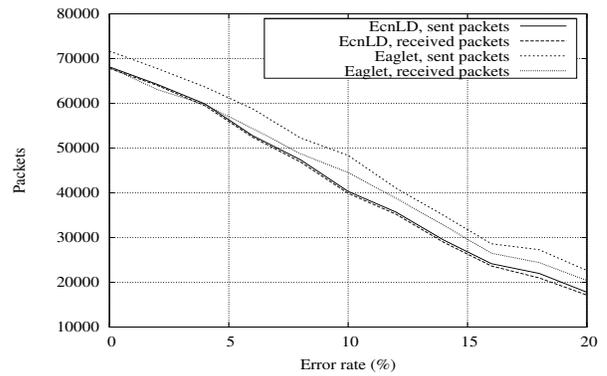


Fig. 8. EcnLD vs Eaglet with competition on a wireless network of 54Mb/s and one retransmission MAC.

EcnLD and TCP-Eaglet in this simulation under the same conditions as above.

a) *Scenario with competition on a wireless network of 11Mb/s:* The results of classification are shown in figures 6 and 7. First, in case of single retransmission, TCP-Eaglet is a slightly better especially for wireless error rates greater than 6%. The reason for this EcnLD drawback is that it has taken enough precautions to prevent congestion in the network. Second, in case of two retransmissions, the difference between EcnLD and TCP-Eaglet seems to be negligible. However, other simulations show that TCP-Eaglet can lead to overestimation of the bandwidth, as shown in the next scenario.

b) *Scenario with competition on a wireless network of 54Mb/s:* This test is designed to show that ECN alone is not enough to differentiate between losses and to make sure that the sender does not overestimate the bandwidth by transmitting packets more than the network can accept. The capacity of wireless networks is set to 54Mb/s in this simulation in order to have a bottleneck on the wired network (between R1 and R2) which will be shared between several flows. The results of classification between congestion losses and wireless losses show that EcnLD has a high ratio of received/sent packets (see figures 8 and 9). This high performance is due to the fact that EcnLD takes a good precaution to prevent congestion while TCP-Eaglet commits important errors of congestion losses

classification. Even if TCP-Eaglet has a higher number of packet reception, it does so by sending much more packets on the network, hence it has a significant higher number of data losses. In case of EcnLD almost all packets sent are received. In case of two retransmissions, EcnLD and TCP-Eaglet packets reception are approximately the same but TCP-Eaglet loses many sent packets.

c) *Loss classification percentage:* The accuracy of loss classification is a percentage representing the rate of losses

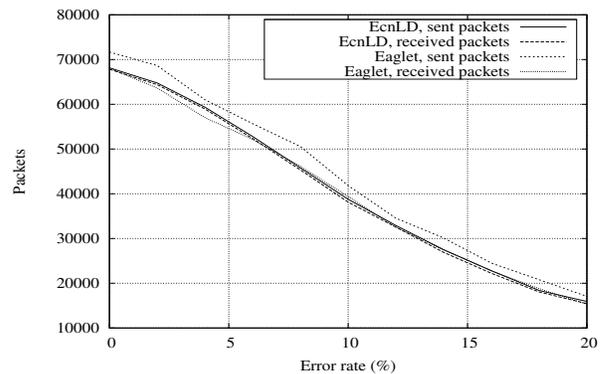


Fig. 9. EcnLD vs Eaglet with competition on a wireless network of 54Mb/s and two MAC retransmissions.

One retrans. 11Mb/s	0%	2%	4%	6%	8%	10%	12%	14%	16%	18%	20%	Avg
EcnLD	100.0%	70.5%	79.4%	61.0%	67.7%	75.5%	68.3%	69.5%	67.3%	68.0%	69.1%	72.39%
Eaglet	100%	59%	84%	84%	86%	92%	90%	92%	89%	87%	88%	86.45%
Two retrans. 11Mb/s	0%	2%	4%	6%	8%	10%	12%	14%	16%	18%	20%	Avg
EcnLD	100.0%	78.3%	76.3%	62.6%	62.9%	78.8%	51.5%	74.1%	71.6%	66.1%	81.0%	73.01%
Eaglet	100%	60%	50%	47%	50%	81%	79%	77%	51%	51%	54%	63.63%
One retrans. 54Mb/s	0%	2%	4%	6%	8%	10%	12%	14%	16%	18%	20%	Avg
EcnLD	40.7%	59.1%	53.2%	62.2%	64.3%	70.7%	75.2%	72.2%	73.4%	65.9%	72.4%	64.48%
Eaglet	3.8%	2.5%	4.1%	4.5%	10.1%	11.9%	26.3%	29.2%	32.8%	29.5%	40.8%	17.77%
Two retrans. 54Mb/s	0%	2%	4%	6%	8%	10%	12%	14%	16%	18%	20%	Avg
EcnLD	40.7%	33.9%	34.3%	43.8%	25.1%	32.5%	65.6%	62.7%	48.2%	38.0%	53.5%	43.48%
Eaglet	3.8%	2.0%	2.2%	2.8%	2.5%	5.9%	9.5%	5.6%	10.5%	8.5%	16.2%	6.31%

TABLE I
ECNLD VS EAGLET WITH COMPETITION (LOSS CLASSIFICATION PERCENTAGE).

properly distinguished from the total number of losses during the simulation. After each simulation run, we look at the simulation trace file to identify the reason for each packet loss, and we compare this with decision taken by the source.

Table I shows the loss classification percentage for EcnLD and TCP-Eaglet; the average percentage for EcnLD is higher in most cases, especially in simulations on the 54Mb/s network.

This explains why in figure 8 TCP-Eaglet has a higher number of packet received but with a lot of lost packets. When EcnLD makes a mis packet classification it comes generally in case of wireless losses. This means that many wireless losses will be classified as congestion losses because of the second condition related to RTT. We take this precaution because on multimedia transmission, it is better to receive most of sent packets than losing them on the network.

V. 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 can be 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 shown that ECN is useful and that it can be used with RTT to overcome the weakness of ECN. We recommend the use of EcnLD for video streaming over wireless networks because the reception rate obtained by EcnLD is very high (the majority of packets are received). We still have tracks to be followed in this particular study to see the effect of the error rate on the increasing of RTT, IAT and ROTT and to evaluate the methods based on these parameters. A next step will be to evaluate the performance of EcnLD for transmission of real video.

REFERENCES

- [1] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 756–769, Dec. 1997.
- [2] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *Wireless Networks*, vol. 1, no. 4, pp. 469–481, Dec. 1995.
- [3] J. Postel, "Transmission control protocol," RFC 793, Sep. 1981.
- [4] S. Biaz and X. Wang, "RED for improving TCP over wireless networks," in *ICWN: International Conference on Wireless Networks*, Jun. 2004, pp. 628–636.
- [5] S. Biaz and N. H. Vaidya, "Discriminating congestion losses from wireless losses using inter-arrival times at the receiver," *IEEE Symposium ASSET*, pp. 10–17, 1999.
- [6] H. Balakrishnan and R. Katz, "Explicit Loss Notification and Wireless Web Performance," in *IEEE GLOBECOM Global Internet*, Nov. 1998.
- [7] J. Postel, "User Datagram Protocol," RFC 768, Aug. 1980.
- [8] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, Internet Engineering Task Force, Jul. 2003.
- [9] E. Kohler, M. Handley, and S. Floyd, "Datagram Congestion Control Protocol (DCCP)," RFC 4340, Mar. 2006.
- [10] S. Floyd and E. Kohler, "Profile for Datagram Congestion Control Protocol (DCCP) congestion control ID 2: TCP-like congestion control," RFC 4341, Mar. 2006.
- [11] S. Floyd, E. Kohler, and J. Padhye, "Profile for Datagram Congestion Control Protocol (DCCP) congestion control ID 3: TCP-friendly rate control (TFRC)," RFC 4342, Mar. 2006.
- [12] S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end differentiation of congestion and wireless losses," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 703–717, Oct. 2003.
- [13] M. K. Park, K.-H. Sohn, and J. H. Jeong, "A statistical method of packet loss type discrimination in wired-wireless network," in *CCNC: IEEE Consumer Communications & Networking Conference*, Jan. 2006.
- [14] Y. Tobe, Y. Tamura, A. Molano, S. Ghosh, and H. Tokuda, "Achieving moderate fairness for UDP flows by path-status classification," in *LCN: IEEE Local Computer Networks*, Aug. 2000, p. 252.
- [15] C.-F. Chou, M.-W. Hsu, and C.-J. Lin, "A trend-loss-density-based differential scheme in wired-cum-wireless networks," in *IWCMC: international conference on Wireless communications and mobile computing*, Jul. 2006, pp. 239–244.
- [16] D. Barman and I. Matta, "Effectiveness of loss labeling in improving TCP performance in wired/wireless networks," Boston University, Tech. Rep., 2002.
- [17] A. Boukerche, G. Jia, and R. W. N. Pazzi, "Performance evaluation of packet loss differentiation algorithms for wireless networks," in *PM2HW2N: Performance monitoring and measurement of heterogeneous wireless and wired networks*. ACM, Oct. 2007, pp. 50–52.
- [18] S. Biaz and X. Wang, "Can ECN be used to differentiate congestion losses from wireless losses?" Auburn University, Tech. Rep. CSSE04-04, 2004.
- [19] G. Ye, T. Saadawi, and M. J. Lee, "Improving stream control transmission protocol performance over lossy links," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 4, pp. 727–736, 2004.
- [20] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP," RFC 3168, Sep. 2001.
- [21] "The Network Simulator NS-2 (v2.31)," <http://www.isi.edu/nsnam/ns/>.
- [22] N.-E. Mattsson, "A DCCP module for ns-2," Master's thesis, Luleå University of Technology, May 2004.