TCP Optimization in Wireless Mesh Backhaul Networks

Sara Grilli Colombo, Tommaso Mandorino, Giovanni Paltenghi
CEFRIEL-Politecnico di Milano
Milan, Italy

***Abstract*— This paper deals with the analysis of cross-layer approaches to TCP performance enhancement in wireless multi-hop networks. The work has been carried out within the MEMBRANE project, which aims to design and develop a reconfigurable multi-antenna, multi-hop wireless backhaul network.**

We considered three TCP algorithms, i.e. NewReno, Jersey and TIBET, and three cross-layering information exchange mechanisms, i.e Explicit Congestion Notification (ECN), Explicit Loss Notification (ELN), Explicit Rate Notification (ERN), the last one expressly devised for the MEMBRANE scenario.

Simulation results show that by adding cross-layering notifications TCP performance can be considerably improved: when the channel quality is high, ERN can offer some improvement to base TCP algorithms performance; when the channel quality is poor, ELN significantly improves performance.

***Keywords*— TCP algorithms, wireless channels, cross-layering, goodput and fairness performance.**

I. Introduction

The problem of performance degradation of traditional TCP protocols when used in wireless networks is well known [1]. The main cause of performance degradation is the high Bit Error Rate (BER) typical of wireless links and the consequent possibility of packet loss due to channel errors, rather than intermediate node buffer overflow. In this case, traditional TCP algorithms consider each packet loss as the result of a network overload, misinterpreting channel errors as congestion events, and inappropriately reduce the congestion window decreasing network throughput. Furthermore, TCP ACK packets are also affected by channel errors and can be lost or delayed, causing round-trip time (RTT) variability and spurious retransmissions.

This paper aims to describe some approaches that we studied and analyzed for TCP performance improvement in the MEMBRANE network. MEMBRANE network [2] is a reconfigurable multi-antenna, multi-hop wireless backhaul network. Much research has been focused on mechanisms to improve TCP performance in one-hop wireless systems, but in wireless multi-hop networks there are further issues that greatly affect TCP performance, due to interaction between MAC, IP and TCP layers [3, 4] that has to be taken into consideration.

Given the MEMBRANE scenario, and in particular the fact that MEMBRANE MAC is TDM/TDMA based, the problem to deal with for TCP optimization is principally the high BER characterizing wireless channels.

In MEMBRANE, a cross-layer approach is used throughout the project: PHY, MAC and routing levels are able to exchange information, so that the resource allocation scheduling and the routing paths assignment are decisions taken with awareness of the network and wireless channel conditions. Leveraging this characteristic, also our work in TCP performance enhancement is focused on cross-layering approaches.

II. TCP Optimization in MEMBRANE Network

In our research for TCP optimization we started from the traditional TCP NewReno protocol, suitable to deal with burst errors, a typical situation of the wireless environment.

We then took into account the approach of using bandwidth estimation to control the congestion window and enhance TCP performance in wireless channel. The considered bandwidth estimation algorithms have been TCP Jersey [5] and TIBET [6]. TCP Jersey joins bandwidth estimation with explicit congestion warning, provided by marking packets flowing through an overloaded path. TIBET does not use congestion notifications from the network layer, but its bandwidth estimation algorithm is more complex than Jersey one.

As said, our work was focused on the possibility to add cross-layer information to improve performance of the selected TCP algorithms. We have identified essentially three possible types of cross-layering mechanisms:

- ECN (Explicit Congestion Notification), as implemented in TCP Jersey: IP notifies TCP of the congestion level of the network, information extracted from the measure of the queue length in the intermediate node buffers.

  If a TCP receiver finds out that the sender has transmitted the segment along a congested path, it echoes this information to the TCP sender via ACK; at the reception of this notification, the TCP sender starts the congestion control procedure.

- ELN (Explicit Loss Notification), as described in [7]: TCP is informed by MAC about each specific packet loss due to channel errors, exploiting the MAC capacity to understand which connection the lost packet refers to, and forward the wrong packet to the recipient.

  The idea is to inform the TCP sender about which packets have been lost due to channel errors, so that it

978-1-4244-1653-0/08/$25.00 ©2008 IEEE ISWPC 2008

will retransmit only lost packets without reducing the congestion window: TCP congestion control mechanism can be decoupled from the retransmission mechanism and set to react only to congestion related loss.

- Notification of the actual available rate on the path (we call it ERN – Explicit Rate Notification), so that the TCP does not transmit more data than the amount the network can actually tolerate. This new approach directly derives from the MEMBRANE framework: since routing and scheduling layers exchange information about the bandwidth that a hop can offer (so that the routing level can take decisions in an aware manner), the proposed idea is to allow also TCP to know this information and use it for limiting the congestion window, therefore the amount of traffic that the sender inject in the network.

III. Simulation Results

In order to evaluate devised cross-layering approaches, we developed a suitable model of the MEMBRANE network in the OPNET simulation environment. Each wireless node implemented in our simulator comprised:

- a standard IP protocol with static routing tables,
- a TDM/TDMA MAC with round robin scheduling,
- a simple PHY layer with variable BER and bit-rate typical of the MEMBRANE wireless.

On top of the IP layer, we added the TCP protocols under investigation. Simulations were carried out comparing the selected "base" versions of TCP (Reno, NewReno, Jersey and TIBET) and considering the three identified cross-layering notifications. First of all we compared all the base versions of the TCP congestion control algorithms, then we considered the addition of notifications.

The analysis was conducted in a simplified MEMBRANE network scenario in which workstations and servers are exchanging file requests and responses with file transfers; the scenario involves intermediate nodes that constitutes the backhaul network which interconnects clients and servers.

Clients and servers exchange data according to an OPNET custom application model: the application model is similar to the standard FTP (File Transfer Protocol), with a client request 256 bytes long, followed by a File Transfer from the server to the client; the size of the transfer is 15 Mbytes. Client and server are about 10 kilometres apart.

The simulation aims to evaluate the performance of TCP congestion control algorithms versus network PER (Packet Error Rate). The chosen measure for TCP performance is goodput, i.e. the number of useful bits per unit of time forwarded by the network from a certain source address to a certain destination, excluding protocol overhead, and excluding retransmitted data packets.

The analysis started from the simplest scenario: a single hop scenario with a single connection, characterized by the absence of intermediate nodes. The second scenario considered was a multi-hop chain (4 hops), with only one traffic flow: with this scenario we can investigate the multi-hop environment and the consequences that it has on the performance. The third scenario considered was a mesh grid network (with 9 intermediate nodes and 6 client-server pairs), with six simultaneous flows that share the network and meet in the intermediate nodes. This is a scenario close to the actual MEMBRANE network. Simulation results described in the following refer to this scenario.

Fig. 1 shows simulation results for the 4 "base" TCP protocols considered, without cross-layering information. Figure 2 shows a zoom of the high PER area of Fig. 1.

As showed in figure, NewReno and Jesey demonstrate the best performance.

NewReno performs well especially at medium-low PER, because of its aggressive behaviour (it enters Fast Retransmit and Fast Recovery phases only one for a burst of errors). When PER is high, its aggressiveness causes network overload and timeouts and performance worsen.

TIBET performs bad at low PER, we believe this is caused by its conservative bandwidth estimation algorithm. In fact, TIBET uses RTTmin (the minimum value of experienced RTT) to calculate the congestion window (CW = BwE x RTTmin, where CW is the congestion window and BwE is the estimated bandwidth), and RTTmin is low especially at low PER. TIBET performance improves at high PER, because RTTmin increase and we have a better estimation of actual available bandwidth.

Jersey uses the current RTT (no RTTmin) to calculate its congestion window. Therefore obtained congestion window value is higher compared to the TIBET one, since the BW x RTT product is higher. This helps Jersey to have better performance. Moreover, ECN helps Jersey to not enter Fast Retransmit and Recovery phases when an error occurs, and this increase its performance at high PER.

Reno shows bad performance at high PER, due to continuous halving of its congestion window for channel errors (continuous entering in Fast Recovery phase).

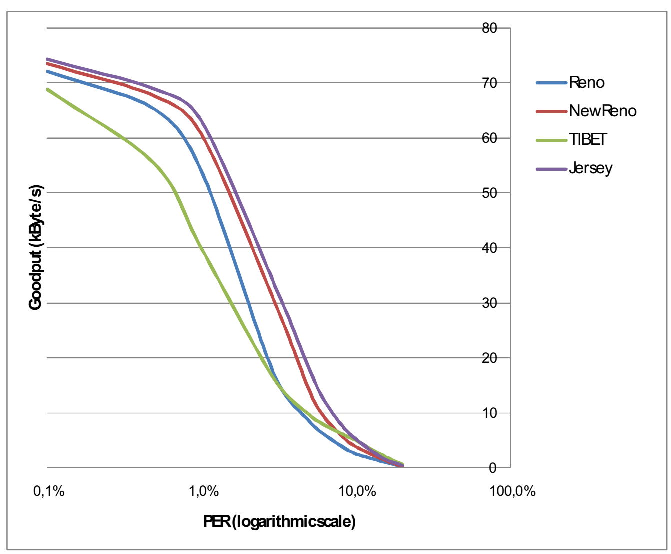

Figure 1. Goodput performance of the considered 4 TCP protocols.

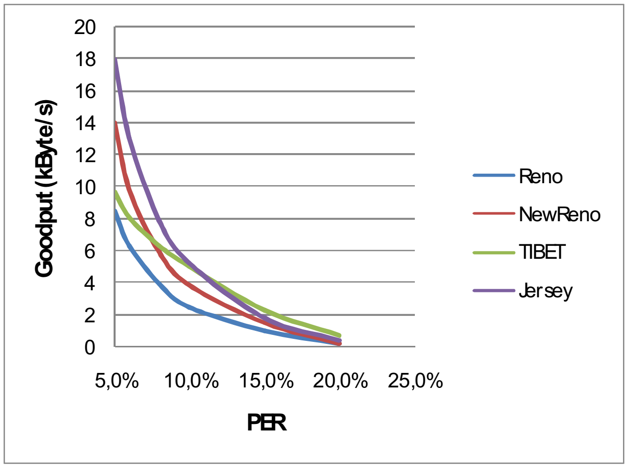

Figure 2. Goodput performance of the considered 4 TCP protocols at high value of PER.

We also measured fairness of the 4 considered TCP protocols, which is how fair TCP protocols share bandwidth among competitive traffic flows. We used Jain's index [8] to measure fairness (Jain's index ranges from 0, no fairness, to 1, optimal fairness).

Fig. 3 shows obtained results for fairness. When channel is good, bandwidth is enough for all the competitive traffic flows and there is no difference in fairness results for TCP protocols. When channel is bad, available bandwidth decrease and traffic flows start to fight for the scarce resource. In particular, NewReno aggressiveness involves low value of fairness at high PER since, when one traffic flow experience a timeout and reduce its congestion window, the other flows are ready to steal its share of bandwidth.

Fig. 4 shows NewReno performance when using the cross-layering mechanisms previously explained.

ERN shows the best performance at low PER: exact knowledge of the available bandwidth helps inject into the network the correct amount of traffic (setting correctly the congestion window) and avoid overloads and timeouts. Aggressive NewReno behaviour manages the few packet losses experienced at low PER.

ELN has the optimum characteristic of reacting very fast to packet losses due to channel errors, this leads to the huge performance improvement over other protocols and cross-layer mechanisms at high PER shown in Fig. 4. At low PER, congestion is dominant and ELN is too aggressive and does not react properly to network overload and congestion.

Fig. 5 shows fairness results for NewReno with cross-layering information. We obtained results similar to the ones analyzed for "base" NewReno: at high PER, NewReno aggressiveness implies low fairness. The difference is for NewReno with ELN that, since it is able to assure enough bandwidth even at high PER, has high fairness over all PER range.

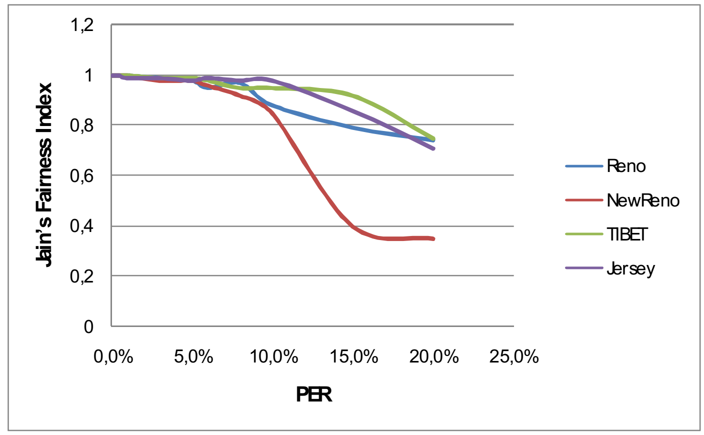

Figure 3. Fairness results of the considered 4 TCP protocols.

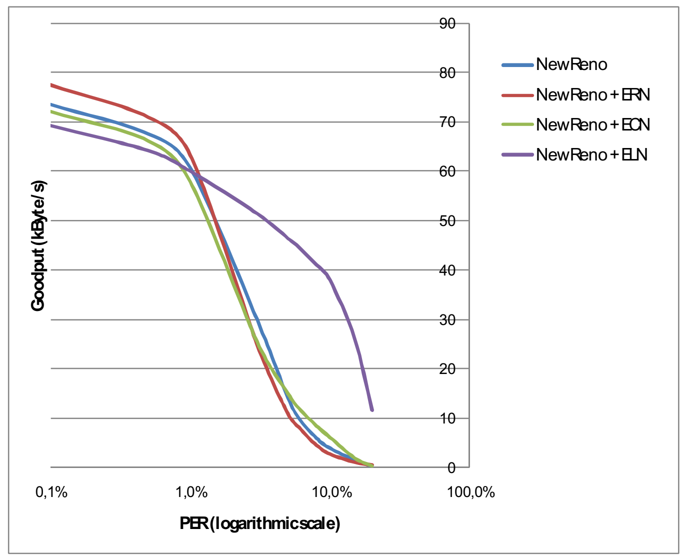

Figure 4. Goodput performance of NewReno with cross-layering information.

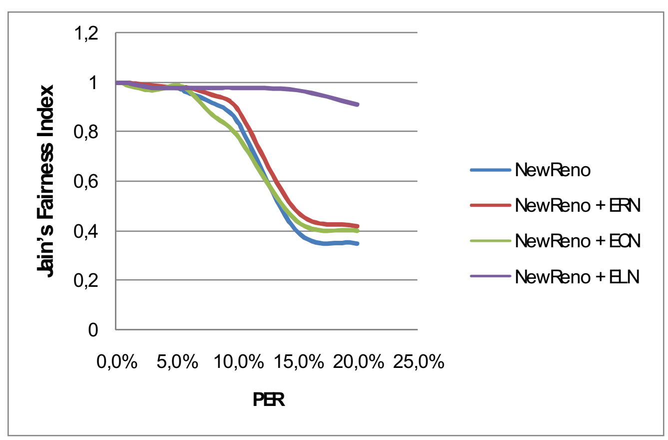

Figure 5. Fairness results of NewReno with cross-layering information.

Finally, Fig. 6 shows the best performing TCP algorithms. From the figure it is clear how there is a threshold value of PER, placed around 1%, under which ERN can offer some improvements to base NewReno performance, but ELN performs badly, because of its aggressive behaviour that burdens the network, without exploiting its feature of fast reaction to packet losses due to channel errors. On the other hand, above 1% of PER, ELN has much better performance respect to the other algorithms and notifications, because it provides a fast and precise information about the main issues that occur for those PER values and properly reacts.

This has led us to study and propose an adaptive TCP algorithm for the MEMBRANE network. Our adaptive algorithm is based on the NewReno protocol. BER/PER has to be passed from lower layers to the adaptive TCP NewReno. According to the current PER value, adaptive TCP NewReno decides which approach to use, the ELN or the ERN:

- if PER is lower than 1%, ERN approach is used,
- if PER is higher than 1%, ELN approach is used.

We are currently in the process of simulating the adaptive algorithm in order to verify its performance in the implemented MEMBRANE network. We are also evaluating the possibility to estimate the exact value of the crossing point between ERN and ELN (the 1% threshold) and not have it fixed in the algorithm, in fact this value depends on network topology.

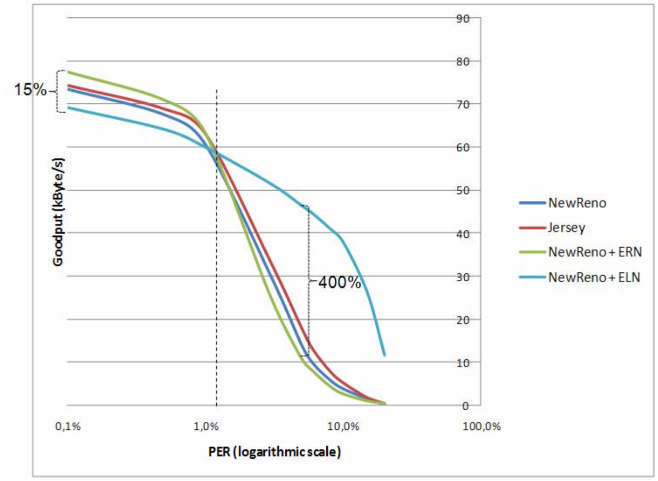

Figure 6. Comparison of the best performing TCP protocols.

IV. CONCLUSIONS

This paper dealt with the analysis of TCP performance enhancement in wireless multi-hop networks, in particular within MEMBRANE project scenarios.

From this initial simulation campaign, the TCP layer performance in a MEMBRANE network can be improved according to the channel quality. The TCP algorithm that performs better is NewReno, together with cross-layer information: when the channel quality is rather high ERN approach can offer some improvements over base TCP algorithms and when the channel quality is poor ELN approach considerably improves average goodput.

This behaviour led us to study and propose and adaptive TCP protocol, that is currently under investigation.

ACKNOWLEDGMENT

The work presented in this paper has been performed in the framework of the IST-MEMBRANE project funded by the European Commission (FP6).

REFERENCES

[1] Y.Tian, K. Xu and N. Ansari, "TCP in wireless environments: problems and solutions", IEEE Communication Magazine, vol. 43, Issue 3, March 2005, pp. S27 – S32

[2] IST MEMBRANE Annex I "Description of work"

[3] Z. Fu, H.. Luo, P. Zerfos, S. Lu, L. Zhang and M. Gerla, "The impact of multi-hop wireless channel on TCP performance". IEEE Transactions on Mobile Computing, vol. 4, issue 2, March-April 2005, pp. 209 – 221

[4] S. Xu, T. Saadawi and M. Lee, "On TCP over wireless multi-hop networks", IEEE Military Communications Conference, 2001. MILCOM 2001, vol. 1, 28-31 Oct. 2001, pp. 282 – 288

[5] K. Xu, Y. Tian and N. Ansari, "TCP-Jersey for wireless IP communications", IEEE Journal on Selected Areas in Communications, vol. 22, issue 4, May 2004, pp. 747 – 756

[6] A. Capone, L. Fratta and F. Martignon, "Bandwidth estimation schemes for TCP over wireless networks", IEEE Transactions on Mobile Computing, vol. 3, issue 2, April-June 2004, pp.129 – 14

[7] B. Zhang, M.N. Shirazi, "Implementation of Explicit Wireless Loss Notification using MAC-layer information", IEEE Wireless Communications and Networking 2003, vol. 2, 16-20 March 2003, pp. 1339 - 1343

[8] R. Jain, D. Chiu and W. Haweu, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system". DEC Research Report, TR-301, September, 1984.