TCP behavior over a greened network

Alessio Sassu
University of Rome "La Sapienza"
Rome, Italy
alessio.sassu@gmail.com

Carmelo Scarso
University of Rome "La Sapienza"
Rome, Italy
carmelo.scarso@gmail.com

Francesca Cuomo
DIET
University of Rome "La Sapienza"
Rome, Italy
francesca.cuomo@uniroma1.it

***Abstract*—Recent works analyze the energy saving improvements of greened networks. The main idea is to define algorithms able to put some network links or devices into sleep mode for saving the used energy. This can achieved by adopting power-aware traffic engineering to route traffic towards a subset of links or to consider topology-based and traffic-based pruning mechanisms. These approaches are very promising as for the energy saving and the overall network performance remain satisfactory, however the question is: are the per-flow traffic performance still guaranteed? How does TCP perform in such a greened network?**
In this paper we analyze the performance of different versions of TCP in a network where energy savings are achieved by the use of one of the algorithm, namely ESTOP, defined to prune the IP network on the basis of network topological characteristics. The performance results show that the behavior of that TCP versions depending on the round trip time, namely *Reno* and *Cubic*, are affected by the link pruning in an expected way. This may allow to use these pruning techniques with a suitable control on the per-flow performance. The tradeoff between energy saving and TCP throughput loss is analyzed.

***Index Terms*—Energy saving, TCP performance, Green Internet**

I. Introduction

Current networking solutions are always becoming more concerned about the energy saving problem, that implies a trade-off between power usage and network performance [1]. The basic principle of most of the solutions proposed to save energy in the Internet is to put some network links into sleep mode on the basis of power-aware traffic engineering methods able to route traffic towards a subset of links [2] [3], to topology-based pruning mechanisms [4] or to traffic based ones [5]. These approaches are very promising as for the energy saving and the overall network performance remain satisfactory, however the question is: are the per-flow traffic performance still guaranteed? How does TCP perform in such a greened network?
As most of the today's Internet traffic is carried by TCP, measuring and analyzing network performance obtained when this protocol is adopted in a greened network is a prime concern.
Since different versions of TCP coexist in the current Internet [6], we focus our attention on their performance after topology changes occur due to the activation and usage of an energy saving procedure.
We conducted a simulation-based study by using the well known NS-2 network simulator [7] and by implementing, in a network adopting Open Shortest Path First (OSPF) for routing, the ESTOP pruning mechanism [4]. Our preliminary analysis shows that even though network performances are obviously affected by the dynamics determined by the switching off of some links induced by the energy saving protocols, it is still possible to achieve good network behaviors when appropriate TCP versions are chosen.
It is to be noticed that a key difference with respect to other TCP analysis under topology changes is that in the ESTOP algorithm we shut down a multiplicity of links at the same time. This is different with respect to a failure scenario when the probability of having multiple, contemporary, switched off links is very low.

The rest of the paper is organized as follows. Section II describes the case study at hand; we consider a current network topology and we run a protocol able to achieve an energy saving by switching off some suitable links. Section III reports the preliminary simulation results. Finally, we conclude the paper in Section IV.

II. Problem definition and case study

Different approaches have been defined in the network energy saving problem framework. Some of them take into account the current traffic load, while others only consider the underlying network topology.
Our study is focused on a solution named ESTOP [4] which belongs to the latter category. In particular, ESTOP switches off links in increasing order of a relevant centrality measure of the network, referred as betweenness, which counts the number of shortest paths (calculated by the use of Dijkstra's algorithm) that include the considered link. The number of links that are switched off by the ESTOP algorithm depends on an input parameter which guarantees that the network connectivity is maintained above a given threshold (we use $50\% - 90\%$ of the initial connectivity degree). ESTOP performance analysis has shown that it is able to save a big portion of network energy by suitably selecting the links that can be switched off.
By following the guidelines of the paper [4] we implemented ESTOP by having it run in every network router and by changing the topology after links switch off by using the classical link state update of OSPF. As a case

978-3-901882-46-3 ©2012 IFIP

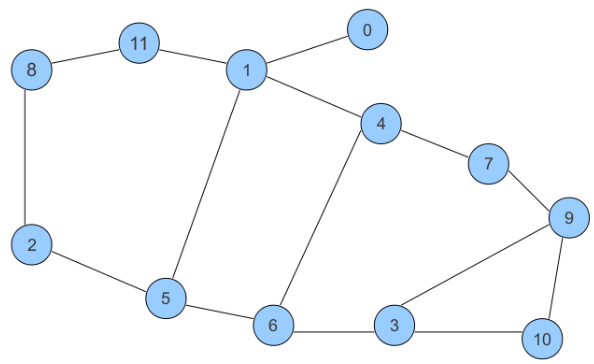


Fig. 1. ABILENE network topology

study we run ESTOP in the ABILENE [8] network case, which presents the network parameters of Table I.

ABILENE is an actual American service provider. Due to the reduced network size we were able to simulate on it realistic traffic scenarios and TCP connections.
Figure 1 shows the network topology where we numbered the different nodes. Nodes in the ABILENE correspond to cities. Propagation delays chosen for all network links refer to the actual distances between the corresponding cities. They vary in the range of 1 *ms* to 11 *ms*.

A TCP flow was established between relatively distant nodes in the network and to simulate a real traffic scenario, we added an uniform background traffic involving all links. On this TCP flow of fixed Segment Size (1040 bytes), we established a FTP file transfer session, where a 1 GB file is downloaded from node 10. The file size has been chosen in order to have a TCP traffic flow for the whole duration of the simulation.
The background traffic is a constant bit rate (CBR) traffic on top of UDP using 25% of the link capacity of every link, before the links switch-off.
Table II reports all simulation parameters, including those related to ESTOP algorithm.

To make our simulations more manageable, we have chosen to scale down by a factor of 1000 both traffic loads and link capacities with respect to the current ABILENE. As can be noted from Table II, we chose as TCP flow endpoints nodes 8 and 10 which represent a worst-case scenario in our topology.

Name	**Value**
Node count	12
Link count	15
Link type	Bidirectional
Minimum degree	1
Average degree	2.5
Maximum degree	4

TABLE I
ABILENE NETWORK PARAMETERS

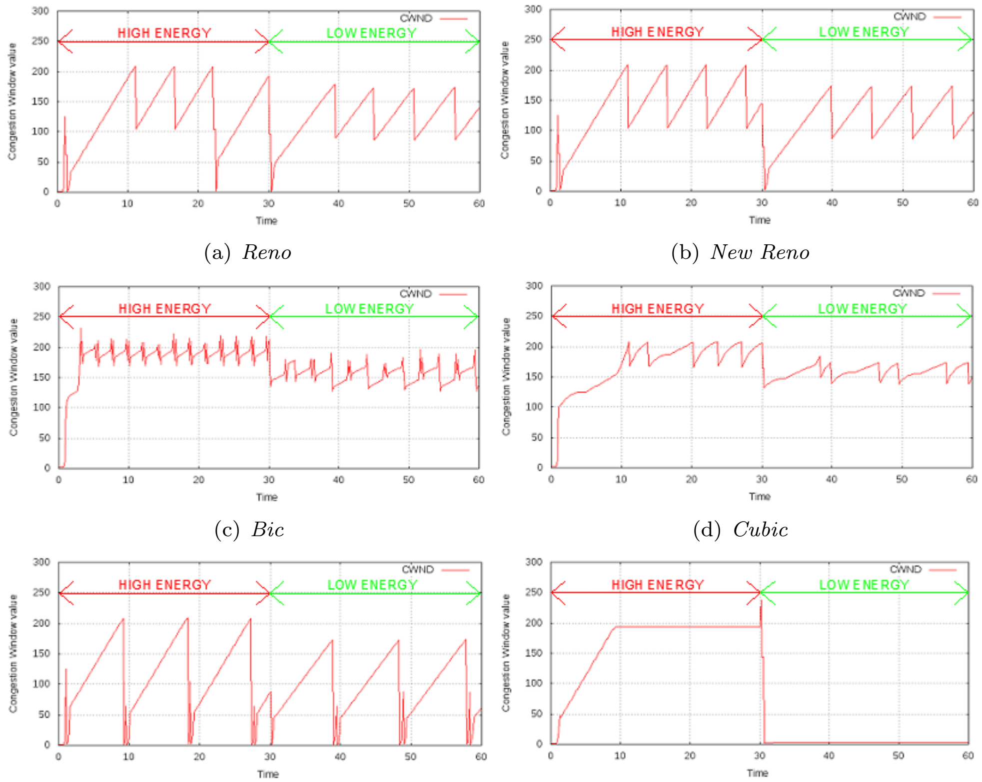
Parameter	**Value**
Link capacity	40 Mbit/s
CBR traffic link usage (percentage)	25 %
TCP flow source node identifier	8
TCP flow destination node identifier	10
Initial source-destination shortest path length (hops)	5
Simulation duration (seconds)	60
ESTOP activation time (seconds)	30
Transition period (seconds)	0.03

TABLE II
SIMULATION PARAMETERS

This is due to two main reasons: i) the selected nodes are, as noted before, relatively distant; ii) for these two nodes, the TCP flow is particularly affected, in terms of available capacity after the links switch-off.
As far as routing is concerned, a OSPF-like link-state protocol has been used. To achieve a sort of worst case we run ESTOP contemporaneously on all routers. This means that the action of putting into sleep mode all the links identified by ESTOP is done by routers at the same time. After ESTOP activation, the network experiences a transition period of a relatively short duration, during which the shortest paths among all nodes are re-calculated and new routing information are propagated thought Link State Advertisement (LSA) in the network. After the transition period, the shortest path between nodes 8 and 10 is increased, in terms of length, by a factor of 1 hop. As explained in [4], the network must remain connected even after the links switch-off. For this purpose, ESTOP algorithm defines a connectivity threshold that in our experiments has been set in the range $50\% - 90\%$.
As far as node features are concerned, we have set their buffer size to be higher than the bandwidth-delay product. In our simulations, different versions of TCP have been tested in the aforesaid setting [6]. Table III reports the different versions of TCP considered in our study. Usage percentages of the different TCP versions refer to real-world settings, as explained in [9].

Name	**Usage**	**Based on**	**Notes**
AIMD	25%	Loss	Includes *Tahoe*, *Reno* and *New Reno*
Cubic	30%	Loss	Used by default in Linux kernels since 2.6.19
Bic	14%	Loss	Used by default in Linux kernels 2.6.8 through 2.6.18
Vegas	2%	Delay	Delay-based modified version of *Reno*
Others	29%	Loss and Delay	There exists several TCP versions suitably modified by the users. It includes CTCP, a TCP version used in Microsoft OS

TABLE III
ANALYZED TCP VERSIONS



(a) *Reno* (b) *New Reno*

(c) *Bic* (d) *Cubic*

(e) *Tahoe* (f) *Vegas*

Fig. 2. Congestion Window values in different versions of TCP.

III. Simulation results

As performance measures, we evaluated both the TCP connection congestion windows size (CWND, expressed in number of segments) and the average throughput (expressed in Mbit/s), since these two parameters accurately reflect network performance and dynamics.
We also measured the percentage of *throughput loss*, defined as the percentage decrease of the average throughput after the ESTOP activation.
The Figure 2 shows the effects of the network dynamics induced by the ESTOP algorithm in terms of congestion window values in our simulations. These behaviors have been derived by setting a connectivity threshold value in ESTOP of 70%. In the figure we indicate as HIGH energy the period before ESTOP activation and the LOW energy the subsequent period.
The common feature of plots in Figure 2 is that after the activation of the ESTOP algorithm, all TCP versions show on average a decreased congestion window value.
This is due to the fact that background UDP flows are rerouted on different network after the ESTOP activation, thus decreasing the residual capacity for the TCP flow, which in turns generates more congestion.
The increased congestion results in lower congestion window average and peak values.
Since the activation of the algorithm increases the path length by one hop, which in turn implies an higher Round Trip Time value, the time needed to reach the congestion window peak value is higher than before the ESTOP activation. Due to being a delay-based TCP version, as will be discussed in the following, this is not true for *Vegas*. As can be noticed in Figures 2(a), 2(b) and 2(e), the AIMD class protocols show a relatively regular behavior reacting quickly to the topology change induced by ESTOP.
For all the three protocols belonging to this class, there exists a significant reduction of the congestion window values limited to a small fraction of time, i.e., the time needed by the network to react to the links switch-off. After this transition time they go to another stable state with decreased congestion window values with respect to the time prior to activation.

Figures 2(c) and 2(d) refer instead to *Bic* and *Cubic* versions. They show lower congestion window variations when loss are detected with respect to the protocols analyzed so far. A difference that there exists with the AIMD class protocols is that both *Bic* and *Cubic* react quicker to the link switch-off and never set their congestion window to one segment.

Figure 2(f) shows the behavior of *Vegas*. *Vegas* keeps the congestion window to a constant value and changes it only when delay variations are detected. Since ESTOP causes an higher Round Trip Time value, *Vegas* reacts decreasing

the congestion window values. The drop is significantly larger than in the loss-based protocol version.
As will be more obvious when the average TCP connection throughput is considered, delay-based protocols like *Vegas* are not suitable for networks where energy saving procedures involving network dynamics are used.
Table IV reports the average TCP connection throughput values for the analyzed protocol versions before and after ESTOP activation, and the relevant throughput loss in performance between the two periods.

We now analyze the behavior in terms of average throughput of the loss-based protocols.
An obvious fact from Table IV is that *Bic* and *Cubic* protocols achieve the best performance in this scenario both before and after ESTOP execution. As previously explained, this behavior is due to the fact that even during the transition period their congestion window values never reach the value of one segment, and in general their peak values are higher.
In particular, *Cubic* is less aggressive and more systematic than *Bic*, and it tries to increase the throughput by following the shape of a cubic function. In particular it starts by following a concave growth where it probes for more bandwidth and proceeds by allowing the network to stabilize before entering a convex growth period in which it allows the throughput to grow quicker than in *Bic*.
This is the reason why *Cubic* achieves the best performance among all protocol versions, after links have been switched off.
With respect to *Tahoe*, *Reno* and *New Reno* achieve better performance both in terms of throughput and in terms of percentage throughput loss thanks to the Fast Recovery mechanism they implement. This is true before and after the use of ESTOP algorithm and implies an higher bandwidth usage.
Different considerations can be done for the *Vegas* protocol version. The throughput loss in this case is the highest one and it reflects the relevant throughput loss in the congestion window value.
As far as the connectivity threshold is concerned, Table III shows that choosing a high threshold (90%), which implies switching off a low number of links, the network is less affected by congestion and can therefore guarantee a higher residual capacity for the TCP connection.
A detailed comparison among the analyzed TCP versions, in terms of percentage average throughput loss is shown in Figure 3.

From Figure 3 we can notice that all protocols show a decreased mean throughput loss when the connectivity threshold is increased. It should also be noted that both *Reno* and *Cubic* achieve the best performance as they experience a lower throughput loss with respect to the other TCP versions.

Special comments can be done for the behavior of TCP *Vegas*, shown in Figure 4.

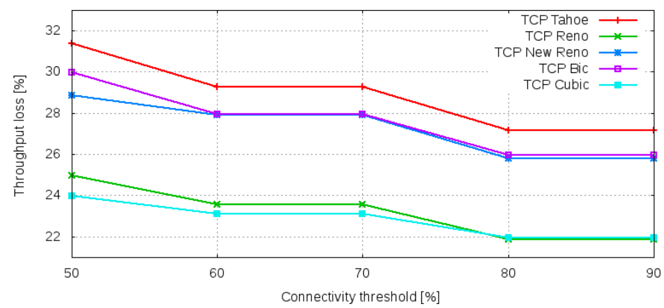


Fig. 3. Throughput loss comparison in different TCP versions

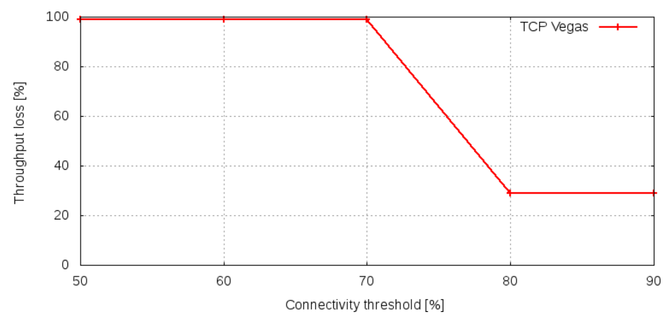


Fig. 4. Throughput loss in TCP *Vegas*

Figure 4 shows that, as previously discussed with respect to *Vegas* behavior in terms of congestion window, *Vegas* can not be used effectively when a sharp topology change happens, as the one induced by the use of energy saving protocols. This is due to the fact that the delay estimated by *Vegas* is not coherent with the actual one after ESTOP. Instead, as can be observed when the connectivity threshold is up to 80%, topology changes that do not imply major delay variations also do not affect excessively the mean throughput.
The throughput loss analyzed so far allows us to state that, especially when some TCP versions are used, is it possible to save energy in the network without a drastic compromise of the TCP performance.
Table V shows the percentage energy saved and the switched off links, with respect to the connectivity threshold of the ESTOP algorithm.

Threshold	**Energy Saving**	**Switched off links**
50 %	26.7 %	(9;10), (9;3), (2;8), (1;5)
60 %	20.0 %	(9;10), (9;3), (2;8)
70 %	20.0 %	(9;10), (9;3), (2;8)
80 %	13.3 %	(9;10), (9;3)
90 %	13.3 %	(9;10), (9;3)

TABLE V
Energy savings percentages and switched off links with respect to connectivity threshold

A Cisco ONS 15600 Switching Platform [10] averages a 140 $Watt$ power consumption for each 2.5 *Gbps* line card. In our scenario, and since we scaled down by a factor of 1000 all link capacities, we can state that using the ESTOP

Version	Before ESTOP	After ESTOP 50%	After ESTOP 70%	After ESTOP 90%
Tahoe	18.11 [Mbit/s]	12.42 [Mbit/s]	12.81 [Mbit/s]	13.19 [Mbit/s]
Reno	20.82 [Mbit/s]	15.62 [Mbit/s]	15.91 [Mbit/s]	16.27 [Mbit/s]
New Reno	21.51 [Mbit/s]	15.30 [Mbit/s]	15.51 [Mbit/s]	15.96 [Mbit/s]
Bic	27.23 [Mbit/s]	19.07 [Mbit/s]	19.62 [Mbit/s]	20.16 [Mbit/s]
Cubic	26.26 [Mbit/s]	19.96 [Mbit/s]	20.19 [Mbit/s]	20.49 [Mbit/s]
Vegas	26.72 [Mbit/s]	0.1 [Mbit/s]	0.32 [Mbit/s]	19.02 [Mbit/s]

TABLE IV
Average throughput values in different versions of TCP

algorithm on the ABILENE topology where the original link capacities are used, can give a energy saving ranging from 9072 $Watt$ (when the 50% value is chosen for the connectivity threshold) down to 4368 $Watt$ (when we set the threshold to 80% or 90%).

IV. Conclusion

This paper presented a preliminary and partial analysis to the evaluation of TCP performance over a greened network. Since there exists a strong correlation between network performance and energy savings, a trade-off between these two parameters must be accurately chosen. For this purpose, a very detailed analysis is necessary for understanding how it is possible to reduce energy consumption without majorally affecting the network performance.

Concerning this issue, we have studied the behavior of various TCP protocol versions, analyzing the congestion window and average throughput values of a TCP connection.

Simulation results demonstrate that there exist a deterioration of performance for all TCP versions, but carefully choosing the appropriate protocol version energy savings can be achieved while maintaining a good network performance. In our scenario we have showed that *Cubic* achieves the best performance in terms of average throughput and features a decrease of only roughly 23.1% with respect to the initial value, when the connectivity threshold is set to 70 %.

We also showed that delay-based versions like *Vegas* are not adequate for this framework where the activation of energy saving procedures causes sensible delay variations. Note that our simulation results refer to an adverse scenario, where both the background traffic rate and the connectivity threshold of the ESTOP algorithm have been chosen such that the residual capacity of the bottleneck link is particularly affected by the network dynamics.

For this purpose, further studies could be focused on testing TCP protocol versions where different settings can be chosen. For instance:

- different background traffic distributions can be used, in particular it can be decreased so that the residual capacity of bottleneck link increases;
- buffer size at nodes can be set equal to the bandwidth-delay product;
- various TCP connections can be simultaneously established in the network to study the interactions among them;
- various TCP versions can be tested on different and bigger topologies;
- both CWND and average throughput can be averaged over all the possible source/destination pairs, to measure the performance gap between the considered worst case with respect to the average case.

Acknowledgment

Authors would like to to thank Fabio Taglialatela for the very constructive discussions on the characteristics of the different TCP versions. Marco Polverini and Antonio Cianfrani helped us to implement in the correct way the ESTOP protocol.

References

[1] R. Bolla, R. Bruschi, A. Carrega, and F. Davoli, "Green network technologies and the art of trading-off," in *IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS*, 2011.

[2] M. Zhang, C. Yi, B. Liu, and B. Zhang, in *2010 18th IEEE International Conference on Network Protocols (ICNP)*.

[3] A. Coiro, F. Iervini, and M. Listanti, "Distributed and adaptive interface switch off for internet energy saving," in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, 31 2011-aug. 4 2011, pp. 1 –8.

[4] F. Cuomo, A. Cianfrani, M. Polverini, and D. Mangione, "Network pruning for energy saving in the internet," *Computer Networks*, vol. 56, no. 10, pp. 2355 – 2367, 2012, special issue on Green communication networks.

[5] A. Bianzino, L. Chiaraviglio, and M. Mellia, "Distributed algorithms for green IP networks," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, march 2012, pp. 121 –126.

[6] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock, "Host-to-Host Congestion Control for TCP," *Communications Surveys Tutorials, IEEE*, vol. 12, no. 3, pp. 304 –342, quarter 2010.

[7] "The Network Simulator NS-2," http://www.isi.edu/nsnam/ns/.

[8] S. Orlowski, R. Wessaly, M. Pioro, and A. Tomaszewski, "Sndlib 1.0 survivable network design library," *Netw.*, vol. 55, pp. 276–286, May 2010. [Online]. Available: http://dx.doi.org/10.1002/net.v55:3

[9] P. Yang, W. Luo, L. Xu, J. Deogun, and Y. Lu, "TCP Congestion Avoidance Algorithm Identification," in *Proceedings of the 2011 31st International Conference on Distributed Computing Systems*, ser. ICDCS '11, 2011, pp. 310–321.

[10] "Cisco ONS 15600 Datasheet," http://www.cisco.com/warp/public/cc/pd/olpl/ps4533/ps4534/prodlit/index.shtml.