A Transport Layer Approach to Improve Energy Efficiency

Muhammad Usman*, Dzmitry Kliazovich†, Fabrizio Granelli*, Pascal Bouvry†, Piero Castoldi‡
*University of Trento, Trento, Italy
{muhammad.usman, fabrizio.granelli}@unitn.it
†University of Luxembourg, Luxembourg
{dzmitry.kliazovich, pascal.bouvry}@uni.lu
‡Scuola Superiore SantAnna, Pisa, Italy
castoldi@sssup.it

***Abstract*—Incorporating energy efficiency into the design of modern communication systems has become an important area of research. However, while most of the proposed solutions are devoted to making network hardware energy efficient, very few works focus on energy efficiency as a fundamental design parameter of network protocols. This paper proposes an analytical model for energy consumption of TCP which relates energy consumption to protocol operation cycles. Based on this model a number of optimization techniques are proposed to reduce energy consumption of TCP. The experiments, performed using NS2 simulations, demonstrate that energy savings can be as high as 93% for multiple TCP flows.**

***Index Terms*—Green ICT, network protocols, TCP, energy efficiency**

I. Introduction

With increasing concern regarding global climate change, the carbon footprint of ICT has become an important issue in modern world. ICT sector alone is responsible for 2.5% to 4% of the Green House Gases (GHG) emission and 8% to 10% of EUs electricity consumption [1]. The contribution of ICT industry towards global carbon emission is around 2% and is increasing at a compound rate of 6% anually [2]. European Commission estimates that a reduction of 15% to 30% in global carbon emissions is required to prevent the 2 °C increase in global temperature [1]. Along with the ecological impact of ICT, the energy efficient ICT is proved to be a fundamental driver of a robust and sustainable economy [2].

The energy footprints in ICT are mainly due to the communication networks, computers and data centers [3]. Communication networks alone are responsible for 30% to 37% of GHG emission produced by ICT [4]. Bianzino et al. [4] present the taxonomy of the research trends in green networking showing that most of the solutions work on the data link and network layers and only a few approaches focus on transport layer protocols like TCP. Most of these approaches focus on network technologies rather than protocols. For example, the authors in [5] and [6] focus on MAC layer to improve energy efficiency. Irish et al. [7] propose a sleep mode option during TCP operation when server can receive a sleep flag indication from clients. Wang et al. [8] propose a method for estimating computing cost of TCP implementation. TCP, being the most used transport layer protocol [9], has a great potential to save energy in ICT sector.

In this paper, we model analytically energy consumption of network nodes during different protocol operational cycles and propose two modifications to improve energy footprint of TCP on network devices. These modifications include (a) adding sleep flag in TCP header and (b) reducing inter-packet gaps (RIPG) in TCP traffic. During each protocol cycle, TCP transmits bursts of packets and then remains idle waiting for the acknowledgements. The proposed solutions enable network nodes to conserve energy during these inactivity periods. The obtained results show that energy savings can be as high as 93% for multiple TCP flows.

The rest of the paper is organized as follows. Section II describes analytical model for energy consumption of TCP protocol. Section III illustrates proposals to improve energy efficiency of TCP protocol. The obtained analytical and simulation results are presented in Section IV. Finally, Section V concludes the paper providing outlook and future directions for work on this topic.

II. Energy Consumption of TCP

TCP ensures reliable communication between the end systems and is the most widely used transport protocol in the Internet. The reliability is ensured by requiring the receiver to successfully acknowledge the received data. The sender adapts its transmission rate with each acknowledgement to fully utilize capacity of the end-to-end link. During congestion avoidance, TCP conservatively increases the congestion window by one segment every Round Trip Time (RTT) until it detects the first segment loss by receiving three duplicate acknowledgements. This detection causes the TCP to drop its congestion window to $W\ /\ 2$. The interval of congestion window evolving from $W\ /\ 2$ to W is defined as one round of TCP.

Many efforts have been devoted to propose the energy consumption model of TCP. However, currently available models are focused on specific network scenarios only, such as wireless cooperative relaying networks, low-power lossy networks and burst transmission over wireless networks. Instead, the model proposed in this paper is generic and cap-

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

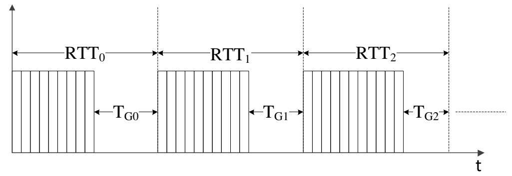

Fig. 1. Gaps between bursts of packets.

tures macroscopic behavior of TCP [10] to account for the length of inactivity periods during each RTT and the energy consumption of network node. During congestion avoidance, the number of TCP packets transmitted in one round can be calculated as follows:

$$\begin{aligned} &\frac{W}{2}+(\frac{W}{2}+\frac{1}{r})+...+(\frac{W}{2}+\frac{N}{r}) \\ &=\sum_{n=0}^{\frac{rW}{2}}(\frac{W}{2}+\frac{n}{r}) \\ &=\frac{3rW^2}{8}+\frac{3W}{4}, \end{aligned} \tag{1}$$

where r is the average number of data packets acknowledged by a single response from the receiver, W is the maximum achievable size of congestion window and N is number of RTTs in one round that is $rW\ /\ 2$.

Eq. (1) can be used to derive the congestion probability P_c, the probability of having one packet loss in one round:

$$P_c=\frac{1}{\frac{3rW^2}{8}+\frac{3W}{4}}=\frac{8}{3rW^2+6W}\ . \tag{2}$$

In addition to the congestion probability P_c, we also consider link error probability P_l, the probability of having one data packet corrupted during its transmission over the network. As a result, the total error probability is given by [11]:

$$P=\frac{P_l}{1-e^{-\frac{P_l}{P_c}}}\ . \tag{3}$$

Assuming independent and uniformly distributed bit errors, packet and bit error rates of the network link can be coupled together as:

$$P_{PER}=1-(1-P_{BER})^M\ , \tag{4}$$

where M is the size of a packet in bits and is equal to $1500\times 8=12000$ bits for the most common Ethernet MTU.

The $3W\ /\ 4$ component in Eq. (2) can be neglected as it is much smaller in comparison with $(3rW^2)\ /\ 8$. Then, the maximum achievable congestion window can be expressed in terms of total error probability P:

$$W=\sqrt{\frac{8}{3rP}}\ . \tag{5}$$

Assuming that P for a given link is known, Eq. (5) can be used to find maximum achievable congestion window W. During inactivity intervals, network switches and routers can be turned to sleep mode to conserve energy. During sleep mode, the network device remains inactive but still listens to the network links for any broadcast or routing related messages. Fig. 1 illustrates general behavior of TCP in the steady (congestion avoidance) state during each round trip time. To ensure reliable communication between end nodes, TCP uses a sliding-window based congestion control, where new segments are transmitted only after acknowledging the reception of previous segments. In the steady state, TCP starts its congestion window, cwnd, with $W\ /\ 2$ and increases it by one, when it receives acknowledgments of all segments sent in last RTT.

Let T_{G_0}, T_{G_1}, and T_{G_n} be the idle intervals of data inactivity, when TCP is waiting for the acknowledgements (and not sending TCP packets).

$$T_{G_0}=RTT_0-(\frac{W}{2})\times\frac{M}{D}[s]\ , \tag{6}$$

$$T_{G_1}=RTT_1-(\frac{W}{2}+\frac{1}{r})\times\frac{M}{D}[s]\ , \tag{7}$$

$$T_{G_n}=RTT_n-(\frac{W}{2}+\frac{n}{r})\times\frac{M}{D}[s]\ , \tag{8}$$

where M is the size of a packet in bits, D is the data transmission rate in bits per second and W is the maximum achievable size of congestion window, given by Eq. (5).

The network device consumes peak power during packet transmission time and idle power during inactivity periods, when it is waiting for the acknowledgements. Eq. (9) provides the energy consumption model of TCP, where we associate power levels to both phases of RTT [12].

$$E_n=T_{G_n}\times P_{Idle}+T_{Tx_n}\times P_{Peak}[J]\ , \tag{9}$$

where P_{Peak} and P_{Idle} are the power consumption of a network node in transmitting and idle modes, n is the RTT number and T_{Tx_n} is the time spent in transmitting packets for the respective RTT.

We estimate the energy consumption of TCP flow during one round by estimating the time spent idle $T_{G_{round}}$ and while transmitting packets $T_{Tx_{round}}$.

$$\begin{aligned} T_{G_{round}}&=\sum_{n=0}^{\frac{rW}{2}}(RTT-(\frac{W}{2}+\frac{n}{r})\times\frac{M}{D}) \\ &=RTT(\frac{rW}{2}+1)-\frac{M}{D}(\frac{3rW^2}{8}+\frac{3W}{4})[s]\ , \end{aligned} \tag{10}$$

$$\begin{aligned} T_{Tx_{round}}&=\sum_{n=0}^{\frac{rW}{2}}(\frac{W}{2}+\frac{n}{r})\times\frac{M}{D} \\ &=\frac{M}{D}(\frac{3rW^2}{8}+\frac{3W}{4})[s]\ . \end{aligned} \tag{11}$$

Eq. (9) is used to measure the energy consumption of TCP flow at a network node by replacing T_{G_n} and T_{Tx_n} by $T_{G_{round}}$ and $T_{Tx_{round}}$, respectively.

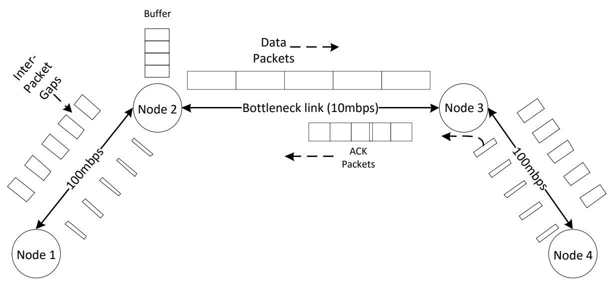

Fig. 2. Inter-packet gaps due to bottleneck links.

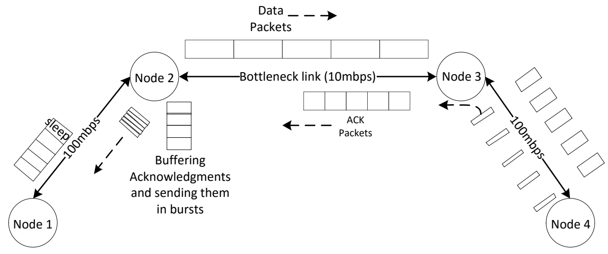

Fig. 3. Buffering acknowledgments at node 2 to combine inter-packet gaps.

The acknowledgement flow was not modeled explicitly, as data and acknowledgement segments are always processed together during active phase of each RTT. The sender continues sending the next congestion window when it receives acknowledgements of the last RTT. For this, the proposed model is based on active and waiting intervals, while T_{G_n} in Eq. (9) represents both data and acknowledgement segments.

III. Proposals to Improve Energy Efficiency

According to the energy consumption model of TCP presented in (9) of Section II, the gaps between packet bursts and size of these gaps define energy footprint of the TCP on the network devices along the data path. The energy consumption peaks during data transmission and can be just a fraction of it during idle intervals.

In steady state, TCP begins each round with congestion window equal to around half of the bandwidth-delay product (BDP) and spends considerable amount of time waiting for the acknowledgements. The amount of waiting decreases as congestion window evolves and approaches the BDP. The packets begin to queue in the buffer and then start to be dropped when buffer becomes full - triggering window reduction at the sender node. During idle intervals, network nodes remain operational and waste energy. These idle intervals constitute a great potential for energy saving. In the following, we propose two solutions to utilize these idle intervals for reducing energy consumption of TCP.

Sleep Flag in TCP header: An availability of explicit signaling can facilitate power management in network devices that take constant time to transition to the sleep mode and wake up. The sender node can include sleep flag in the optional field of TCP header of the last packet in the transmitted burst, indicating estimated duration of inactivity period in milliseconds. This sleep flag marks the beginning of inactivity interval during which network node can switch to low power mode and conserve energy until the next burst of packets is received. Eq. (8) provides an estimation of sleep interval based on the estimation of TCP round trip time with time stamps that might not be precise, but is accurate enough to decide whether it worth initiating the sleep mode or if the inactivity interval is smaller than transitioning time. The transitioning time of a typical network interface is estimated to be under 0.5 milliseconds [13].

Energy consumption of network device in processing a TCP flow increases proportionally to RTT. High RTTs result in waste of energy during inactivity intervals. Putting network devices/interface cards to sleep during these intervals can save considerable amount of energy. The proposed technique is inspired by the results obtained in [7] enhanced with flow aggregation (FA) technique.

Multiple TCP flows require flow aggregation to enable sleep mode option during packet inactivity intervals. The flows are synchronized in a way that their congestion window starts at the same time. This synchronization is achieved by buffering data segments of TCP flows of smaller RTTs to wait for the beginning of active interval of longer-RTTs TCP flows.

Reducing Inter-Packet Gaps (RIPG): Bottleneck links create dispersion between packets traveling on the end-to-end path [14]. Slower link increases the time of each packet transmission. Then packet transmission time becomes smaller at the faster subsequent links, which creates gaps between the packets (see Fig. 2). These gaps travel along the path and arrive at the source with the acknowledgements from the receiver. The source node generates one data packet for every received acknowledgement. Therefore, inter-packet gaps created by the bottleneck link tend to propagate into subsequent packets of the TCP flow. This way, consolidating inter-packet gaps can save significant amount of energy.

It is observed that, for larger RTTs inter-packet gaps can be above 1 ms, which is well above the transitioning time of a network interface [13]. Combining inter-packet gaps (see Fig. 3) and enlarging sleep intervals becomes a feasible solution to save energy. For example, buffering 16 data packets can consolidate a sleep interval of 15 ms, if the bottleneck link creates 1 ms time gaps between packets. For multiple TCP flows, flow aggregation and RIPG are used together to save energy on underlying nodes.

We investigated the impact of MAC layer protocols on the proposed techniques and found that MAC protocols for fixed wired networks (e.g., Ethernet LAN) do not affect the suggested transport layer techniques. The end hosts are usually connected to an Ethernet switch using unshielded twisted pair cable (UTP), which separates uplink, so the media is no longer shared. The hosts transmit on one and receive on a different UTP. On the other hand, in wireless LAN (WLAN),

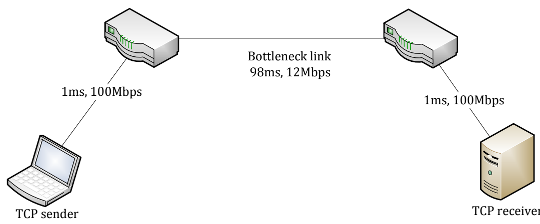

Fig. 4. Simulation scenario.

the dynamic buffer sizing [15] and access point buffering [16] techniques can be implemented to handle the large bursts of data generated by the flow aggregation and maintaining high throughput and low delay across the network.

IV. Evaluation

This section presents performance evaluation of the proposed solutions. Fig. 4 details the simulation scenario with a bottleneck link of 12 Mbps. The buffer limit of bottleneck link is 50 packets. A data file of 10 MB is transferred from TCP sender to the receiver using multiple TCP flows. The time difference d between RTTs of these flows is kept as 5 ms, 10 ms, 15 ms, 20 ms and 25 ms for different simulation scenarios. For each data file, energy consumed by the sender node and average TCP throughput are observed. The simulations are performed using network simulator NS2[1]. For all simulation scenarios, network nodes consume 300 W during the peak load, 212 W when they are idle and 1 W when in the sleep mode.

Sleep Flag in TCP header: Fig. 5 compares energy consumption of TCP and savings obtained with a sleep flag and RIPG methods. TCP flows are aggregated to synchronize the beginning of each congestion window and transmit data segments of all flows in a single burst. It can be observed that in standard TCP the energy consumption increases with increasing value of d and m, where d is the time difference between RTTs of multiple TCP flows and m is the total number of TCP flows. A network node consuming 2603 joules with a single flow begins to consume 15694 joules for six TCP flows, when RTTs of these flows have 5 ms difference from each other. The same network node consumes 22670 joules of energy for 25 ms difference in RTTs of these flows. Introduction of the sleep flag in TCP header brings improvement in energy consumption of underlying nodes. For example, the energy footprint of six TCP flows is reduced from 15694 joules to 12093 joules for $d = 5\ ms$ and from 22670 joules to 17436 joules for $d = 25\ ms$. This corresponds to 23% energy savings in both cases. This energy saving comes on the expense of degradation in average TCP throughput. Fig. 6 presents the average throughput degradation due to flow aggregation and RIPG. The throughput degradation is proportional to d and m. Larger values of m and d render higher percentages of throughput degradation. TCP throughput for six flows is decreased from 9.16 Mbps to 8.64 Mbps in case of $d = 5\ ms$ and from 7.61 Mbps to 5.98 Mbps for $d = 25\ ms$.

The detailed simulation results are presented in Table I. It compares percentages of energy savings and the consequent average throughput degradation for different values of d and m. When multiple TCP flows are transmitted together on the same link, the bandwidth is shared among multiple flows so the average TCP throughput is decreased, as shown in Table I. Introduction of the sleep flag option does not alter TCP throughput dynamics, as it affects only protocol inactivity period. However, when multiple TCP flows are aligned to synchronize active and waiting intervals, the flows with shorter RTTs buffer their data segments and wait for the larger RTTs flows to receive the acknowledgments and start next transmission round together. This deceases the overall TCP throughput, as shown in Fig. 6 and Table I. It can be observed that the overall energy saving ranges from 21% to 25% for multiple TCP flows, when flow aggregation and sleep mode options are both enabled. The average throughput degradation ranges from 1.82% to 21.41%. The throughput degradation depends on the time difference between the RTTs of multiple TCP flows. The lesser is the difference the larger is the number of TCP flows that can be synchronized together without any significant impact on TCP throughput. For example, when 200 ms flow is synchronized with 205 ms flow with sleep flag option enabled in TCP header, 24.61% energy can be saved with only 1.82% of throughput degradation. However, when same TCP flow is synchronized with 225 ms TCP flow, same percentage of energy saving is observed with 5.88% of throughput degradation. Similar trend is observed for more than two flows.

Reducing Inter-Packet Gaps (RIPG): The amount of energy saved by the reduction of inter-packet gaps is more than that obtained from sleep flag and flow aggregation as shown in Fig. 5. The amount of the saved energy is proportional to d and m. Larger values of d and m leads to higher percentages of energy saving at the expense of higher throughput degradation. For multiple flows, the techniques of flow aggregation and RIPG can be used together. Six TCP flows with 25 ms of RTT difference provide 93.28% of energy saving and 28.53% of throughput degradation. A network node consuming 15694 joules of energy begins to consume only 1496 joules if inter-packet gaps are merged together. Similar percentages of energy savings are observed for other values of m and d. The throughput degradation is also proportional to the values m and d. Smaller values of d give lower values of throughput degradation. For example, throughput degradation for $d = 5\ ms$ ranges from 1.82% to 5.71% with 89.77% to 90.47% energy savings. Smaller-RTT flows are affected more with throughput degradation, as they need to wait for longer-RTT flows to start their congestion window. Larger values of d lead to larger waiting times and higher is the throughput degradation. For example in case of 25 ms RTT difference between six TCP flows, the average throughput is degraded from 7.60 Mbps to 5.44 Mbps.

[1]http://www.isi.edu/nsnam/ns/

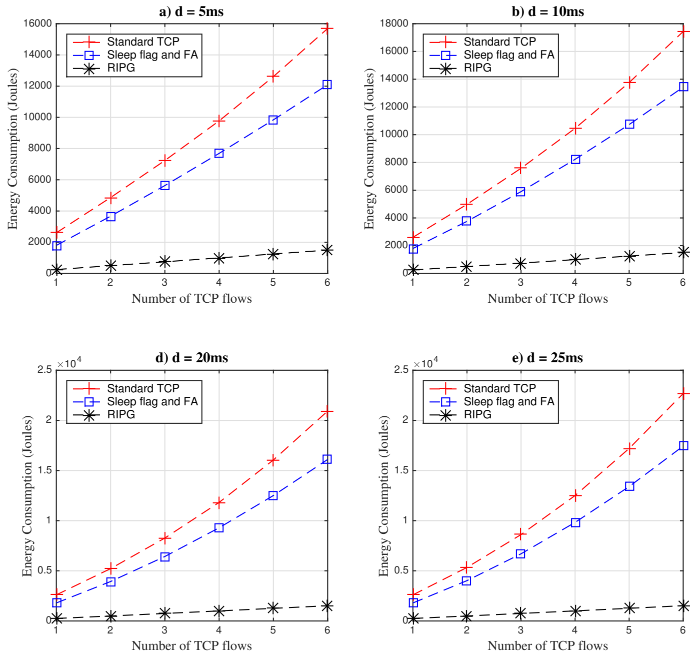

Fig. 5. Impact of sleep flag, flow aggregation and RIPG on energy consumption of network node.

TABLE I
PERCENTAGES OF ENERGY SAVINGS AND THROUGHPUT DEGRADATION FOR SLEEP FLAG AND FA

	Energy Savings					Throughput Degradation				
TCP Flows (m)	**d=5ms**	**d=10ms**	**d=15ms**	**d=20ms**	**d=25ms**	**d=5ms**	**d=10ms**	**d=15ms**	**d=20ms**	**d=25ms**
2	24.61	24.62	24.63	24.61	24.61	1.82	2.39	5.67	4.76	5.88
3	22.18	22.20	22.23	22.19	22.18	2.72	4.69	6.83	8.84	10.74
4	21.31	21.35	21.38	21.33	21.31	3.56	6.77	9.70	12.38	14.84
5	21.96	22.00	22.03	21.98	21.96	4.65	8.71	12.30	15.49	18.36
6	22.95	22.99	23.03	22.95	22.95	5.71	10.52	14.65	18.24	21.41

TABLE II
PERCENTAGES OF ENERGY SAVINGS AND THROUGHPUT DEGRADATION FOR RIPG

	Energy Savings					Throughput Degradation				
TCP Flows (m)	**d=5ms**	**d=10ms**	**d=15ms**	**d=20ms**	**d=25ms**	**d=5ms**	**d=10ms**	**d=15ms**	**d=20ms**	**d=25ms**
2	89.77	90.01	90.23	90.44	90.65	11.60	12.89	14.22	13.21	14.07
3	89.69	90.14	90.55	90.93	91.28	11.37	13.11	15.28	17.05	18.74
4	89.82	90.46	91.02	91.52	91.96	12.18	15.24	17.83	20.20	22.54
5	90.13	90.92	91.59	92.16	92.66	13.20	16.90	20.16	23.07	25.68
6	90.47	91.38	92.13	92.75	93.28	14.20	18.75	22.31	25.76	28.53

Table II provides the detailed results of energy saving and throughput degradation for multiple TCP flows. It can be observed that the energy-saving percentages are persistent for different values of RTT. This is due to the fact that, in all the cases, the network node transmits a burst of packets and all smaller gaps are merged together to switch the network node to sleep for inactivity intervals. In conclusion, the percentage of energy saving remains proportional to the number of TCP flows. However, energy saving comes at the expense of TCP throughput which is dropped due to the waiting time of shorter-RTT flows to synchronize with larger-RTT flows. In case of mixed flows with shorter and longer RTTs, the large-RTT flows are synchronized together, while the small-RTT flows can be synchronized with burst transmission of buffered packets in RIPG technique. In this way, a considerable amount of energy can be saved without significant impact on average TCP throughput.

V. CONCLUSION AND FUTURE WORK

The paper presents a mathematical model for the energy consumption of TCP and two novel solutions to improve energy efficiency. The proposed solutions take advantage of the

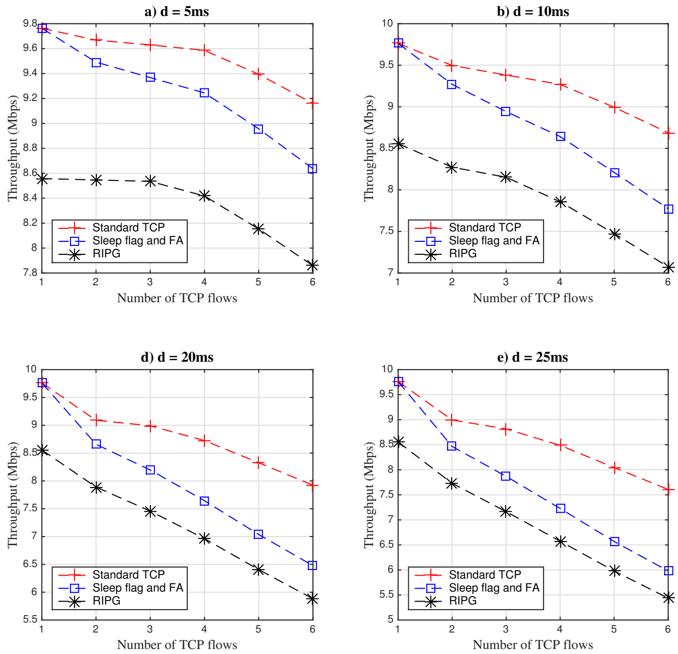

Fig. 6. Impact of sleep flag, flow aggregation and RIPG on average throughput of network node.

bursty nature of TCP flows and enable sleep mode in network nodes to save power during protocol inactivity periods. The achieved energy savings can be as high as 92% for a single TCP flow and up to 93% for multiple TCP flows. For multiple TCP flows, improvement in energy consumptions comes at the expense of TCP throughput degradation, which was observed to be as high as 29%.

Future work will focus on analysis of the delay considerations of the proposed scheme and its adaptation to delay sensitive applications, such as VOIP and remote control.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the funding from National Research Fund, Luxembourg in the framework of ECO-CLOUD and iShOP projects.

REFERENCES

[1] Neelie Kroes, "Greening the Digital World: Major Companies to Measure the Environmental Footprint of ICT," *[Online]. Available: http://ec.europa.eu/archives/commission_2010-2014/kroes/en/blog/ict-footprint.html*, published: 2nd March 2012, visited on: 30th October 2015.

[2] J. Laitner, "The Energy Efficiency Benefits and the Economic Imperative of ICT-Enabled Systems," *ICT Innovations for Sustainability, ser. Advances in Intelligent Systems and Computing, Eds. Springer International Publishing*, vol. 310, pp. 37–48, 2015.

[3] K. Bilal, S. Khan, S. Madani, K. Hayat, M. Khan, N. Min-Allah, J. Kolodziej, L. Wang, S. Zeadally, and D. Chen, "A survey on Green Communications Using Adaptive Link Rate," *Cluster Computing*, vol. 16, no. 3, pp. 575–589, 2013.

[4] A. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, "A Survey of Green Networking Research," *IEEE Communications Surveys Tutorials*, vol. 14, no. 1, pp. 3–20, January 2012.

[5] A. Antonopoulos and C. Verikoukis, "Multi-player game theoretic mac strategies for energy efficient data dissemination," *IEEE Transactions on Wireless Communications*, vol. 13, no. 2, pp. 592–603, February 2014.

[6] V. Miliotis, L. Alonso, and C. Verikoukis, "Energy efficient proportionally fair uplink offloading for ip flow mobility," *IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 6–10, December 2014.

[7] L. Irish and K. Christensen, "A Green TCP/IP to Reduce Electricity Consumed by Computers," *Proceedings IEEE. Southeastcon '98*, pp. 302–305, April 1998.

[8] B. Wang and S. Singh, "Computational energy cost of TCP," *Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies. INFOCOM 04*, vol. 2, pp. 785–795, March 2004.

[9] S. Hagag and A. El-sayed, "Enhanced TCP Westwood Congestion Avoidance Mechanism (TCP WestwoodNew)," *International Journal of Computer Applications*, vol. 45, no. 5, pp. 21–29, May 2012.

[10] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," *SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 3, pp. 67–82, 1997.

[11] J. Mata-Diaz, J. Alins, J. Munoz, and O. Esparza, "A Simple Closed-Form Approximation for the Packet Loss Rate of a TCP Connection Over Wireless Links," *IEEE Communications Letters*, vol. 18, no. 9, pp. 1595–1598, September 2014.

[12] H. Singh and S. Singh, "Energy Consumption of TCP Reno, Newreno, and SACK in Multi-hop Wireless Networks," *SIGMETRICS Perform. Eval. Rev.*, vol. 30, no. 1, pp. 206–216, 2002.

[13] M. Gupta and S. Singh, "Dynamic Ethernet Link Shutdown for Energy Conservation on Ethernet Links," *IEEE International Conference on Communications. ICC '07*, pp. 6156–6161, June 2007.

[14] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, "CapProbe: A Simple and Accurate Capacity Estimation Technique," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 67–78, 2004.

[15] T. Li, D. Leith, and D. Malone, "Buffer Sizing for 802.11-Based Networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 1, pp. 156–169, February 2011.

[16] E. Tan, L. Guo, S. Chen, and X. Zhang, "PSM-throttling: Minimizing Energy Consumption for Bulk Data Communications in WLANs," *IEEE International Conference on Network Protocols. ICNP '07*, pp. 123–132, October 2007.