



Reinforcement Learning and Energy-Aware Routing

Piotr Fröhlich

pfrohlich@iitis.pl

Institute of Theoretical and Applied
Informatics, IITIS-PAN
Gliwice, Silesia, Poland

Erol Gelenbe

seg@iitis.pl

Institute of Theoretical and Applied
Informatics, IITIS-PAN
Gliwice, Silesia, Poland
& Lab. I3S, Université Côte d'Azur
Nice, France

Mateusz Nowak

mateusz@iitis.pl

Institute of Theoretical and Applied
Informatics, IITIS-PAN
Gliwice, Silesia, Poland

ABSTRACT

We present an approach that uses Reinforcement Learning (RL) with the Random Neural Network (RNN) acting as an adaptive critic, to route traffic in a SDN network, so as to minimize a composite Goal function that includes both packet delay and energy consumption per packet. We directly measure the traffic dependent energy consumption characteristics of the hardware that we use (including energy expended per packet) so as to parametrize the Goal function. The RL based algorithm with the RNN is implemented in a SDN controller that manages a multi-hop network which assigns service requests to specific servers so as to minimize the desired Goal. The overall system's performance is evaluated through experimental measurements of packet delay and energy consumption under different traffic load values, demonstrating the effectiveness of the proposed approach.

CCS CONCEPTS

• **Networks** → **Data path algorithms**; • **Hardware** → **Power estimation and optimization**; **Neural systems**; • **General and reference** → **Experimentation**; **Measurement**; **Performance**; • **Theory of computation** → **Routing and network design problems**; **Online learning algorithms**; **Reinforcement learning**; • **Computing methodologies** → *Modeling and simulation*.

KEYWORDS

smart networks, energy consumption, quality of service, machine learning, reinforcement learning, neural networks, optimization

1 INTRODUCTION

Many recent analyses have pointed to the significant growth in energy consumption by ICT, going from roughly 5% of total electricity production a decade ago [13] to some 8-10% of total electricity produced in more recent years [7, 8, 25]. Some of these estimates may be viewed as being excessively alarmist. Indeed, ICT's energy efficiency is also improving and predictions of the future are sometimes based on simulations from technological characteristics

dating from around 2010 [1]. Indeed, some recent studies actually point to a stabilization of the growth in energy consumption by data centers and network operators [28], and in addition the amount of renewable energy which is used by ICT has also increased.

However the inevitable growth in energy consumption due to the IoT, the increasingly powerful wireless networks at the edge and the increasing variety and size of software that supports the Internet as discussed several years ago [35], are a matter of significant concern. Thus, both the operating cost of ICT's electricity consumption and the concerns regarding the sustainability of the manufacturing and decommissioning of ICT devices and systems, have been discussed in recent government reports in Europe [5].

These concerns have generated early research in networks that select paths with an objective of maximizing network life-time when energy availability is limited [16]. They have resulted in proposals for the design of computer systems and networks that are more energy efficient [4, 31–33], and for communications that operate with renewable energy [15]. They have also given rise to many studies about sustainable computing [40] and energy efficient digital design [30]. Thus in addition to the usual considerations of system cost, performance and Quality of Service (QoS), energy awareness has become a part of system and network management. The lack of accurate measurements regarding power consumption by ICT have also resulted in a suggestion for the use of sensor networks in this area [32].

A relatively static approach to energy savings, such as “putting equipment to sleep” when it is lightly loaded, may appear attractive because of its relative simplicity. However experiments show that it can lead to serious degradation in network Quality of Service (QoS) and increases in packet delay, loss and jitter [20]. Thus early work [19] sought to adaptively obtain improvements in IP network paths to reduce energy consumption without impairing QoS. This work resulted in the design of experimental Energy Aware Routing Protocols using network self-awareness [12, 29, 41], and an early version implemented reinforcement learning (RL) at the routers of an IP network to achieve energy savings [18].

Software defined networks (SDN) [2, 3, 34, 39] offer an opportunity to exploit data about the state of the network for energy optimisation as suggested in [9, 22–24], and machine learning (ML) techniques can optimize Fog system performance including QoS, security [26, 27] and energy [10, 14]. Thus in this paper we present an approach that uses the Random Neural Network (RNN) [11, 21] acting as an adaptive critic, with a Goal based Reinforcement Learning (RL) [38] algorithm to route traffic in a cluster of SDN network nodes, to obtain a compromise between packet delay and energy consumption per packet.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

FlexNets'21, August 27, 2021, Virtual Event, USA

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8634-0/21/08...\$15.00

<https://doi.org/10.1145/3472735.3473390>

This differs from our early work using RL where the RNN was used in each packet router to select the next hop for packets heading to a given destination [17, 36]. In [14], we focused on Security and QoS, while the present paper addresses QoS and Energy consumption. In [10] we used direct current (DC) measurements, after the AC/DC conversion in the power supply, to estimate the *power* consumed in the SDN network, and the optimization then dealt with Power (in the SDN network as a whole) and QoS, the latter being for end-to-end user services. However, the present paper provides measurements of AC power consumption (i.e. root-mean-squared or RMS power) in a network node before the AC/DC converter, so that the *converter's losses are also included*. Furthermore we translate the power measurements into energy (i.e. Joules) rather than power (i.e. Watts) consumption per packet, which is used in our goal function $C(u)$ together with packet delay for each path u .

In Section 2 we present the architecture of the system we consider and detail our energy consumption measurements using an existing energy measurement API, for the hardware units that we use. The purpose of these energy measurements is to introduce relevant parameters in the Goal function of the RL scheme.

Then in Section 3 we summarise the RL algorithm, and present our experimental results that show that Machine Learning with RL is an effective way to operate the system in a manner which minimizes the composite Goal function with QoS and network energy consumption. Finally, conclusions and suggestion for further work are presented in Section 4.

2 SYSTEM DESCRIPTION AND POWER MEASUREMENTS

The system we consider that is shown in Figure 1 consists of seven network nodes (the round objects); the central node acts as the SDN controller while the six other nodes are SDN forwarders. It also includes seven servers (the vertical boxes) connected to the nodes, and whose role is to respond to service requests generated by twenty-four client machines (shown in green) which are attached to all nodes except the central node which is acting as the SDN controller. The client machines generate the traffic of requests to the servers, and the traffic sent by the client machines travels is routed from source to destination over the nodes of the SDN network. The Goal based RL algorithm, which we describe in Section 3, is implemented at the SDN controller

Each network node is an Intel NUC consisting of an Intel(R) Core(TM) i7-8705G CPU @ 3.10GHz processor with 16GB of RAM, running the Ubuntu 18.04.4 LTS Operating System with a Linux Kernel 5.3.0-59-generic (x86-64). Each server is a virtual machine with the previously described configuration running with two cores, 2GB of RAM and the Kali Linux 2020.1 Operating Sstem with the Linux 5.5.0-kali2-amd64 Kernel.

2.1 Power Measurements

It is straightforward to measure the response time to a request generated by an end user, by simply using the time-stamp at the receiving machine for the instant when the request is sent, and the time-stamp of the instant when the response is received. However, measuring the energy consumed is more complicated, and requires

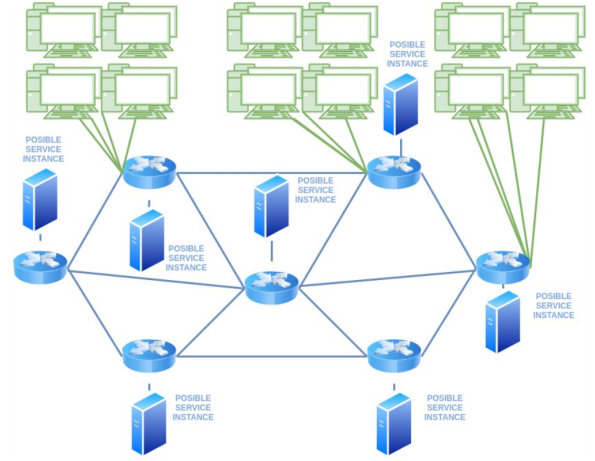


Figure 1: The system architecture used in our experiments, consisting of twenty-four client machines, with a network composed of seven NUC nodes. Seven virtual machines are attached to the nodes and support the services requested by the clients.

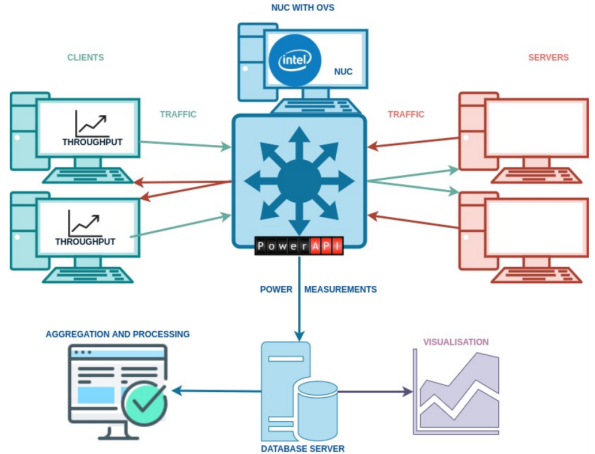


Figure 2: Setup used for measuring the energy consumption (Watts) of an Intel NUC machine.

us to obtain the power (watts) versus traffic load characteristics of the NUC machine that we use through the system.

To this effect, we conducted measurements (in advance of testing our system) for the hardware that we have used, with the setup shown in Figure 2, using the power measurement API internal to the NUC, togetherwith traffic generated by clients and servers connected to the NUC.

The experiments resulted in the data concerning a single NUC used as a forwarded that is summarized in Figure 3 where we show both the type of raw data, in watts versus load in Mb/sec, that was obtained (in light blue) and the smoothed data obtained using a specific Savitzky-Golay filter [37], because of the substantial variability that was observed. The resulting energy (in joules) per packet (shown in blue), versus the traffic rate is shown in Figure

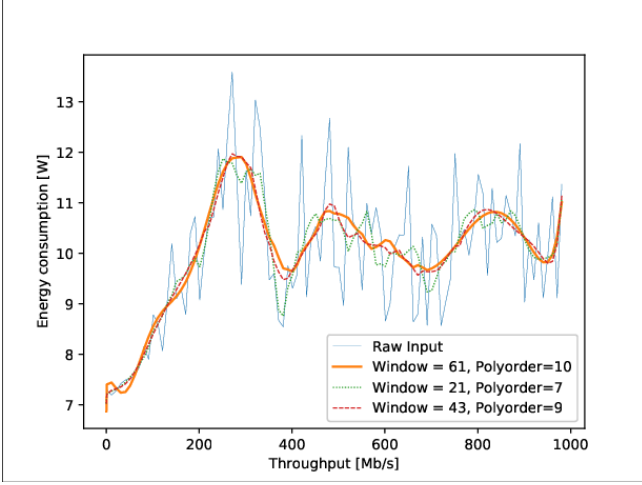


Figure 3: Measurements of the dependence between the load sent through one NUC forwarder and the power used by the NUC forwarder. The measured data has been smoothed with a Savitzky-Golay filter of polynomial order 10 and window parameter 61.

4, together with measurements of delay per request incurred in traversing the node, that is shown in red.

The power values in Figure 3 allow us to readily compute the energy consumed per packet (Joules/packet) by simply dividing the power consumed $P(\tau)$ for a given traffic rate τ by the traffic rate itself, to obtain $E(\tau) = P(\tau)/\tau$ given in the blue curve at the left hand side of Figure 4, normalized with respect to the smallest value of traffic rate at the left hand side of the Figure 3.

These measurements suggest that there is a “green” range of values of workload for the router where both the incurred delay is low, and the energy consumption per packet is also low. Obviously a system control algorithm would attempt to maintain all of the routers in the network in this range of values, whenever possible.

3 REINFORCEMENT LEARNING BASED CONTROL

The objective or Goal function for the system we have described has the form:

$$C(u) = \alpha Q(u) + \beta E(u), \quad (1)$$

where for a given multihop network path u connecting a service request at an end user to the server where the service will be executed, $Q(u)$ is the total packet delay, $E(u)$ is the total energy consumed per packet on that path, and $\alpha, \beta \geq 0$ are the relative weights of the two components of $C(u)$ for the path u . Typically, we take $\alpha = 0.3$ and $\beta = 1$, unless we are considering just QoS or just Energy.

The Cognitive Packet Network (CPN) algorithm [12] used in our earlier work associates a RNN to each network node in order to take decisions regarding the next hop that packets should take. The approach taken here is different: for each source-destination pair (s, d) an RNN chooses among all the paths that can be associated to

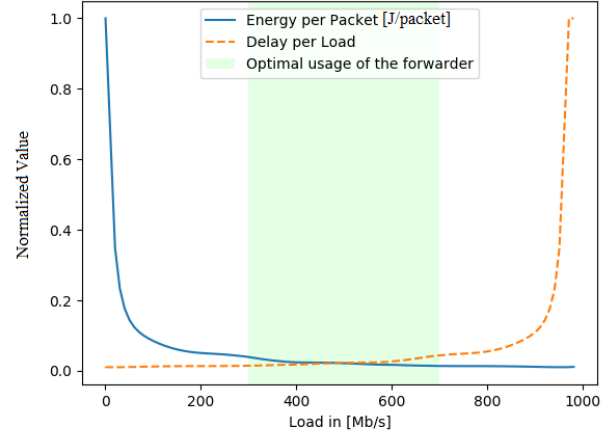


Figure 4: The normalized value of the energy used per packet resulting from the NUC power measurements is shown in blue; these normalized values are with respect to the smallest value of traffic rate at the left hand side of Figure 3. Similarly, the curve in red is the measured delay at the router as a function of its traffic load.

that specific source and destination, because the number is relatively small.

To implement the RL algorithm, the quantity $Q(u)$ is constantly being measured by “smart packets” (SP) travelling over all the active paths u in the SDN network, and collecting successive delays on adjacent links. Once a SP has completed its travel over a given path, it returns the data that it collected to the RL algorithm’s “cognitive memory” (CM), where the measurements are averaged over each successive time window of 100ms. Similarly $E(u)$ is estimated as the sum of the energy per packet using the measured characteristics of a NUC, and from traffic measurements collected by SPs from each node it visits. As a result, the CM contains the data needed to compute $C(u)$ for each u , which is updated as successive SPs bring data to the CM.

Based on this data, the reinforcement learning algorithm computes the “reward”, defined as:

$$R(u) = \frac{1}{C(u)}, \quad (2)$$

for any path u . However, for a source-destination pair (s, d) , there will in general be several paths $u_{s,d}$ starting at s and ending at d that may be used. The “best” one is selected with the help of a RNN [?] specialized in the pair (s, d) so that for each possible path $u_{s,d}$, there is one neuron of this particular RNN. The excitation level of the neuron u for the RNN representing the connection (s, d) is denoted $q_{s,d}^u$.

The historical average of the reward $T_{s,d}^l$ is maintained to represent the time-averaged value of the reward measured for all connections from s to d :

$$T_{s,d}^{l+1} \leftarrow a \cdot T_{s,d}^l + (1 - a) \cdot R(u)|_{u=u_{s,d}}, \quad 0 \leq a \leq 1, \quad (3)$$

where we set $a = 0.9$ to create an average “historical window” of 10 measurements. To select a give $u_{s,d}$ when a new path must be chosen from s to d , the RL first updates the excitatory and inhibitory weights of this specific RNN, so that:

- All excitatory weights reinforcing the current path $u_{s,d}$ are increased if the most recently computed reward value for this path are greater than the historical reward, i.e. $R(u_{s,d}) > T_{s,d}^l$,
- If $R(u_{s,d}) \leq T_{s,d}^l$ then the excitatory weights to all paths other than $u_{s,d}$ are increased,
- And the sum of all weights is kept constant by renormalizing all the weights at each step so that the effect of opposite relative weights is strengthened.

Finally, the new path for the (s, d) pair is chosen simply by computing the $q_{s,d}^u$ for each valid path u from s to d , and choosing the best one:

$$u_{s,d}^* = \arg \max \{q_{s,d}^u : \forall u_{s,d} \text{ from } s \text{ to } d\}, \quad (4)$$

which is the path represented by the most “excited” neuron, i.e. the path whose representative in the neural network has the highest probability of excitation.

3.1 Experimental Results

We have run two sets of experiments, the first with the value $\beta = 0$ in the Goal function, so that the RL algorithm is only seeking to minimize the QoS, and a second set of experiments where we set $\alpha = 0.3$ and $\beta = 1$ so that the RL algorithm is trying to effect a compromise between QoS (expressed as end-to-end delay) and the energy consumption per packet.

In all the experiments, we have varied the overall system load simply by increasing the traffic forwarded by users during their requests to the servers from a total of 1 Gb/sec to 7 Gb/sec. The results are summarized in Figures 5 and 6.

Figure 5 clearly shows that a lower end-to-end delay is obtained, especially in the high traffic ranges, if the RL algorithm just focuses on minimizing delay. However, if the RL algorithm tries to minimize a combination of delay and energy, then delay as a whole will be higher, even at low traffic levels. At low traffic levels, the energy per packet consumption is higher than at high traffic levels, therefore the RL algorithm will sacrifice delay minimization, and end up with higher delay values as shown in Figure 5.

The results presented in Figure 6 are complementary to these observations. We see that when we try just to minimize QoS (i.e. indicated as “delay”, in blue), energy consumption per packet is very high – especially at lower traffic levels, and much more moderate values are achieved if a compromise is achieved – as shown in the red dots – between energy consumption and delay per packet.

All in all, these experiments show that the RL algorithm is responding correctly to the choice of the Goal function, and the results obtained are very promising in this respect.

4 CONCLUSIONS AND FURTHER WORK

In this paper we present a design for implementing RL for minimizing end to end delay and energy consumption, in a SDN controller that selects paths in a SDN network, and adapts efficiently to the

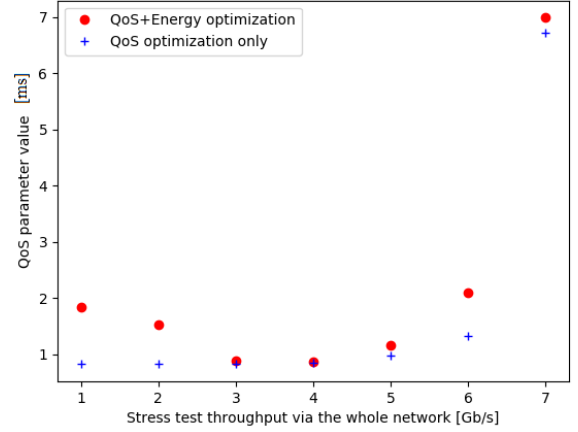


Figure 5: End-to-End Delay in milliseconds averaged over several experiments for each single value of throughput, as a function of overall network traffic throughput shown along the x-axis. We observe that at low traffic rates, the use of energy optimization together with delay minimization, results in higher delay values.

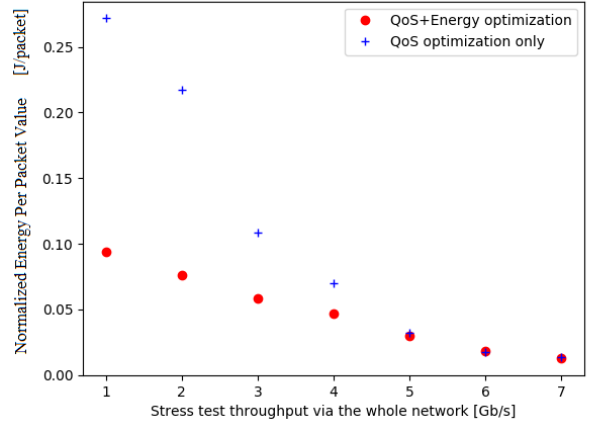


Figure 6: Energy consumption in Joules per packet averaged over several experiments for each value of network throughput, for different values of overall network traffic throughput shown along the x-axis. We observe that the use of only delay minimization, results in much higher energy consumption per packet at low and intermediate values of traffic rates. At high traffic rates, since the network is saturated, the energy consumed per packet reaches its lowest value, but of course the resulting average packet delay is very high as shown in Figure 5.

overall objectives given in the Goal function. However more work will be needed to compare such systems to other approaches and solutions.

In the approach we have presented, RL operates constantly, so that it will be useful to study the case where RL controls are only invoked periodically, for instance each 100, 200 or more milliseconds, so as to reduce the resulting overhead and gather more data.

Another aspect of interest is to compare such AI based techniques with conventional optimization, both in terms of the resulting performance and the computational overhead, and also to factor in the possibility of invoking the controls at different interval lengths as mentioned above.

In future work, it will also be important to evaluate the transient response of such systems as was done in previous work on SDN transients, and better understand whether they can rapidly respond to time-varying changes in workload [6].

Thus the promising approach we present also raises several interesting questions which can be addressed in future research. As adaptive controls – whether optimization based or using AI and ML – become more sophisticated in the Internet, we can expect that network robustness to specific attacks that target the controls themselves will become a major research issue.

ACKNOWLEDGEMENTS

This research is part of the IoTAC (Security By Design IoT Development and Certificate Framework with Front-end Access Control) which introduces multiple innovations to the ISO/IEC CD 30141 IoT Reference Architecture, based on new access control mechanisms that integrate security modules and procedures. It is funded by the European Commission H2020 as a Research and Innovation project under Grant Agreement No. 952684, in the call topic SU-ICT-02-2020 on “Building blocks for resilience in evolving ICT systems”.

REFERENCES

- [1] Anders S. G. Andrae and Tomas Edler. 2015. On Global Electricity Usage of Communication Technology: Trends to 2030. *Challenges* 6, 1 (2015), 117–157. <https://doi.org/10.3390/challe6010117>
- [2] Kamal Benzekki, Abdeslam El Fergougui, and Abdelbaki Elbelrhiti Elalaoui. 2016. Software-defined networking (SDN): A survey. *Security and Communication Networks* 9, 18 (2016), 5803–5833. <https://doi.org/10.1002/sec.173>
- [3] Pankaj Berde, Matteo Gerola, Jonathan Hart, Yuta Higuchi, Masayoshi Kobayashi, Toshio Koide, Bob Lantz, Brian O'Connor, Pavlin Radoslavov, William Snow, and Guru Parulkar. 2014. ONOS: Towards an Open, Distributed SDN OS. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking (Chicago, Illinois, USA) (HotSDN '14)*. ACM, New York, NY, USA, 1–6. <https://doi.org/10.1145/2620728.2620744>
- [4] Andreas Berl et al. 2010. Energy-efficient cloud computing. *The computer journal* 53, 7 (2010), 1045–1051.
- [5] Mireille Campana, Jean Cueniet, Michel Scmitt, and Cédric Siben. 2019. Réduire la consommation énergétique du numérique. *Conseil Général de l'Économie, Ministry of Finances, Paris* (2019). https://www.economie.gouv.fr/files/files/directions_services/cge/consumation-energie-numerique.pdf
- [6] Tadeusz Czachórski, Erol Gelenbe, Godlove Suila Kuaban, and Dariusz Marek. 2021. Time-Dependent Performance of a Multi-Hop Software Defined Network. *Applied Sciences* 11, 6 (2021). <https://doi.org/10.3390/app11062469>
- [7] Maxime Efovi-Hess. 2019. Climate Crisis: The unsustainable use of online video. <https://theshiftproject.org/wp-content/uploads/2019/07/2019-02.pdf>
- [8] Hugues Ferreboeuf and et al. 2019. Lean ICT: Towards Digital Sobriety.
- [9] Frederic Francois and Erol Gelenbe. 2016. Optimizing secure sdn-enabled inter-data centre overlay networks through cognitive routing. In *Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), 2016 IEEE 24th International Symposium on*. IEEE, 283–288.
- [10] P. Fröhlich, E. Gelenbe, J. Fiolka, J. Checinski, M. Nowak, and Z. Filus. 2021. Smart SDN Management of Fog Services to Optimize QoS and Energy. *Sensors* 21 (April 2021), 3105. <https://doi.org/10.3390/s21093105>
- [11] Erol Gelenbe. 1989. Random neural networks with negative and positive signals and product form solution. *Neural Computation* 1, 4 (1989), 502–510.
- [12] Erol Gelenbe. 2009. Steps toward self-aware networks. *Commun. ACM* 52, 7 (2009), 66–75. <https://doi.org/10.1145/1538788.1538809>
- [13] Erol Gelenbe and Yves Caseau. 2015. The impact of information technology on energy consumption and carbon emissions. *Ubiquity* 2015, June (2015), 1–15.
- [14] E. Gelenbe, J. Domanska, P. Fröhlich, M. P. Nowak, and S. Nowak. 2020. Self-Aware Networks That Optimize Security, QoS, and Energy. *Proc. IEEE* 108, 7 (July 2020), 1150–1167. <https://doi.org/10.1109/JPROC.2020.2992559>
- [15] E. Gelenbe, D. Gesbert, D. Gunduz, H. Külah, and E. Uysal-Biyikoglu. 2013. Energy harvesting communication networks: Optimization and demonstration (the E-CROPS project). In *24th Tyrrhenian International Workshop on Digital Communications - Green ICT (TIWDC)*. IEEE Xpress, 1–6. <https://doi.org/10.1109/TIWDC.2013.6664201>
- [16] Erol Gelenbe and Ricardo Lent. 2004. Power-aware ad hoc cognitive packet networks. *Ad Hoc Networks* 2, 3 (2004), 205–216.
- [17] Erol Gelenbe, Ricardo Lent, and Zhiguang Xu. 2001. Design and performance of cognitive packet networks. *Perform. Evaluation* 46, 2-3 (2001), 155–176. [https://doi.org/10.1016/S0166-5316\(01\)00042-6](https://doi.org/10.1016/S0166-5316(01)00042-6)
- [18] Erol Gelenbe and Toktam Mahmoodi. 2011. *Energy-Aware Routing in the Cognitive Packet Network*. 7–12. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.725.5635&rep=rep1&type=pdf>
- [19] E. Gelenbe and S. Silvestri. 2009. Optimisation of Power Consumption in Wired Packet Networks. In *QShine 2009. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Vol. 22. Springer, Berlin, Heidelberg, 717–629. https://doi.org/10.1007/978-3-642-10625-5_45
- [20] E. Gelenbe and S. Silvestri. 2009. Reducing power consumption in wired networks. In *24th International Symposium on Computer and Information Sciences*. IEEE Xpress, 292–297. <https://doi.org/10.1109/ISCIS.2009.5291829>
- [21] Erol Gelenbe and Andreas Stafylopatis. 1991. Global behavior of homogeneous random neural systems. *Applied Mathematical Modelling* 15, 10 (1991), 534–541.
- [22] Frédéric Giroire, Nicolas Huin, Joanna Moulhierac, and Truong Khoa Phan. 2018. Energy-Aware Routing in Software-Defined Network using Compression. *Computer Journal* 61, 10 (2018), 1537–1556. <https://doi.org/10.1093/comjnl/bxy029>
- [23] N. Huin, M. Rifai, F. Giroire, D. Lopez Pacheco, G. Urvoey-Keller, and J. Moulhierac. 2017. Bringing Energy Aware Routing Closer to Reality with SDN Hybrid Networks. (2017), 1–7. <https://doi.org/10.1109/GLOCOM.2017.8254456>
- [24] Nicolas Huin, Myriana Rifai, Frédéric Giroire, Dino Lopez Pacheco, Guillaume Urvoey-Keller, and Joanna Moulhierac. 2018. Bringing Energy Aware Routing Closer to Reality With SDN Hybrid Networks. *IEEE Trans. Green Commun. Netw.* 2, 4 (2018), 1128–1139. <https://doi.org/10.1109/TGCN.2018.2842123>
- [25] Nicola Jones. 2018. How to stop data centres from gobbling up the world's electricity: The energy-efficiency drive at the information factories that serve us Facebook, Google and Bitcoin. *Nature* 561 (September 2018), 163–166. <https://doi.org/10.1038/d41586-018-06610-y>
- [26] Kübra Kalkan, Gürkan Gür, and Fatih Alagöz. 2017. Defense Mechanisms against DDoS Attacks in SDN Environment. *IEEE Communications Magazine* 55, 9 (2017), 175–179.
- [27] Kübra Kalkan and Sherali Zeadally. 2017. Securing Internet of Things (IoT) with Software Defined Networking (SDN). *IEEE Communications Magazine* (November 2017). <https://doi.org/10.1109/MCOM.2017.1700714>
- [28] George Kamiya. 2020. Data Centres and Data Transmission Networks. <https://www.iea.org/reports/data-centres-and-data-transmission-networks>
- [29] Ariane Keller, Daniel Borkmann, Stephan Neuhaus, and Markus Happe. 2014. Self-Awareness in Computer Networks. *Int. J. Reconfig. Comp.* 2014 (2014), 692076:1–692076:16. <https://doi.org/10.1155/2014/692076>
- [30] Santosh K. Kurinec and Sumeet Walia (Eds.). 2020. *Energy Efficient Computing & Electronics: Devices to Systems*. CRC Press. <https://doi.org/10.1109/MIE.2020.2966835>
- [31] Ricardo Lent. 2010. A sensor network to profile the electrical power consumption of computer networks. In *EEE Globecom Workshops*. 1433–1437.
- [32] Ricardo Lent. 2010. Simulating the power consumption of computer networks. In *15th IEEE International Workshop on Computer Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD)*. 96–100.
- [33] Ricardo Lent. 2015. Analysis of an energy proportional data center. *Ad Hoc Networks* 25 (2015), 554–564.
- [34] Rahim Masoudi and Ali Ghaffari. 2016. Software defined networks: A survey. *Journal of Network and Computer Applications* 67 (2016), 1–25. <https://doi.org/10.1016/j.jnca.2016.03.016>
- [35] Barbara Pernici et al. 2012. What IS can do for environmental sustainability: a report from CAISE'11 panel on Green and sustainable IS. *Communications of the Association for Information Systems* 30, 1 (2012).
- [36] Georgia Sakellari. 2010. The cognitive packet network: A survey. *Comput. J.* 53, 3 (2010), 268–279.

- [37] Abraham Savitzky and M. J. E. Golay. 1964. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry* 36, 8 (1964), 1627–1639. <https://doi.org/10.1021/ac60214a047>
- [38] R. S. Sutton and A. G. Barto. 2018. *Reinforcement learning: An Introduction, 2nd edition*. MIT Press.
- [39] Thomas Vachuska, Andrea Campanella, Carmelo Cascone, Charles Chan, Jordan Halterman, Jian Li, and Brian O'Connor. 2020. Home Page of Onos Project: Open source SDN controller. <https://onosproject.org> [Online; Last accessed 13 Mar 2020].
- [40] R. Verdecchia, F. Ricchiuti, A. Hankel, P. Lago, and G. Procaccianti. 2017. Green ICT Research and Challenges. In *Advances and New Trends in Environmental Informatics. Progress in IS*, V. Wohlgemuth, F. Fuchs-Kittowski, and J. Wittmann (Eds.). Springer, Cham. https://doi.org/10.1007/978-3-319-44711-7_4
- [41] Yuanyuan Zeng, Deshi Li, and Athanasios V. Vasilakos. 2013. Real-time data report and task execution in wireless sensor and actuator networks using self-aware mobile actuators. *Comput. Commun.* 36, 9 (2013), 988–997. <https://doi.org/10.1016/j.comcom.2012.07.016>