Gradient optimisation for network power consumption

Erol Gelenbe and Christina Morfopoulou

Dept. of Electrical & Electronic Engineering
Imperial College, London SW7 2BT, UK
e.gelenbe@imperial.ac.uk
c.morfopoulou@imperial.ac.uk

Abstract. The purpose of this paper is to examine how a gradient-based algorithm that minimises a cost function that includes both quality of service (QoS) and power minimisation in wired networks can be used to improve energy savings with respect to shortest-path routing, as well as against a "smart" autonomic algorithm called EARP which uses adaptive reinforcement learning. Comparisons are conducted based on the same test-bed and identical network traffic. We assume that due to the need for network reliability and resilience we are not allowed to turn off routers and link drivers. We also assume that for QoS reasons (notably with regard to jitter and to avoid packet desequencing) we are not allowed to split traffic from the same flow into different paths. Under these assumptions and for the considered traffic, we observe that power consumed with the gradient-optimiser is a few percent to 10% smaller than that consumed using shortest-path routing or EARP. Since the magnitude of the savings is small, this suggests that further power savings may only be obtained if under-utilised equipment can be dynamically put to sleep or turned off.

1 Introduction

Much work has been devoted to power savings in wireless sensor networks where battery power can be crucial, including Topology Control [1, 2, 3] dynamically adjusting radio transmission power and hence range so as to preserve connectivity of each potential source-destination pair. In [4] it is indicated that the radio transceiver, which is the dominant energy consumer within a sensor, consumes almost the same amount of energy in transmit, receive and idle mode. In [5] energy efficient routing for ad hoc networks is presented using the Cognitive Packet Network (CPN) [6, 7] with smart packets that improve QoS and energy savings. A recent comprehensive survey on energy efficiency in mobile networks can be found in [8]. Early work for wired networks [9] proposed traffic aggregation along a few routes, a modification of network topology by route adaptation and putting certain nodes and devices to sleep. A network-wide approach (coordinated sleeping) as well as a link layer approach (uncoordinated sleeping) were discussed and the possible effects on routing protocols were examined. In [10] an energy saving algorithm for Ethernet links using local data to make sleeping decisions was suggested, while powering components on/off in combination with an offline multicommodity network-flow problem for traffic assignment was considered in [11]. An online technique was proposed [12] to spread load through multiple paths, based on a step-like model of power consumption as a function of the hardware's processing rate and the ability of nodes to automatically adjust their operating rate to their utilization. Rate-adaptation for individual links was examined in [13] based on the utilization and the link queuing delay, where traffic is sent out in bursts at the edge routers enabling other line cards to sleep between successive bursts. In [14] the authors select the active links and routers to minimize power consumption via simple heuristics that approximately solve a NP-hard problem. In [15] a case study based on specific backbone networks is discussed, and an estimate of the potential overall energy savings in the Internet is presented in [16]. In [17] the reduction of power consumption in wired networks in the presence of users' QoS constraints and experiments with dynamic traffic management in conjunction with the turning on/off of link drivers and/or routers is discussed, and using the Cognitive Packet Network (CPN) [18] routing protocol for energy awareness in conjunction to QoS is considered. Energy efficiency is examined for Cloud Computing in [19]. Power measurements of network components [11] indicate that the base system is the largest power consumer: it is best to minimize the number of chassis at a given point of presence (PoP) and maximize the number of cards per chassis. In [13] the impact of the hardware processing rate and traffic on power consumption yields measurements similar to those in Figure 1 from [20] for routers.

Here we apply a queuing theory based gradient method described in [21] for QoS and power minimisation in wired networks to improve upon (i) shortest-path routing and (ii) an experimental autonomic algorithm (EARP) [22] for QoS and power optimisation. The algorithm is limited to a single step of the gradient descent in order to provide fast computation, and the algorithm is initiated for (i) with the shortest path algorithm, and for (ii) with EARP. Comparisons are conducted using the same test-bed and the same network traffic. We assume that due to the need for network reliability and resilience we do not turn off routers and link drivers, and that for QoS reasons (notably with regard to jitter and packet desequencing) we do not split traffic from the same flow into different paths. Under these assumptions we observe that the energy consumed using the gradient-optimiser when it is started with known shortest-paths or with paths discovered by EARP is smaller by a few percent, and savings are greater when starting with paths provided by EARP which selects paths based on power optimisation. We note that even a few percent in power savings, scaled up to the power consumed in high-speed routers over long periods of time, can lead to significant economic and carbon dioxide (CO2) savings in energy. The CO2 emissions obviously depend on whether the energy used is from nuclear, fossil, or renewable sources such as hydroelectric, wind or electrovoltaic. However energy savings will have at least the same proportion in CO2 savings, because lower energy use offers greater opportunity to run routers and link drivers using a combination of renewable and stored energy, and there is also less need for cooling using fans and air conditioning equipment.

2 Network Optimisation

Probability models have long been used to design and optimise computer systems and architectures [23]. Here we follow the same tradition by designing an algorithm that is based on a class of probability models for networks called G-networks [24, 25] with multiple classes [26], that were initially inspired by biological neural systems [27]. This mathematical model includes the queues that form at routers and links due to the flow of payload packets, as well as the flow of control packets that are used to re-route traffic for power savings or QoS improvement. For lack of space, we only sketch the optimisation algorithm which can be found elsewhere [21]. The contribution of this paper is to evaluate how this algorithm will improve power consumption over well known shortest path routing and the smart adaptive algorithm EARP. The G-network model allows us to represent the fact that control packets used to optimise are also adding to congestion and power consumption, and the model includes their effect on performance, on the overhead that they induce, and on power consumption.

The network model that is used in the optimisation algorithm model has a set of N queues: router queues R or link queues L, $\mathbf{N} = \mathbf{R} \cup \mathbf{L}$. We use r and l to denote a router or link, $r \in \mathbf{R}$ and $l \in \mathbf{L}$. A traffic class k is a flow of packets between a source-destination pair (s, d), which travels on a path to destination. $\lambda(r, k)$ is the packet rate of user class k arriving from outside the network to router r and $\lambda(r, k) > 0$ only if r is the source node for class k. The probability that a packet of class k travels in one step from node i to node j is $P(i, k, j)$. We also have control traffic classes (r, k) acting on user class k at router r, and $\lambda^{-}(j, (r, k))$ is the rate at which such control packets may enter the network via router i. The probability that a control packet of class (r, k) travels from node i to j in one hop is $p((r, k), i, j)$. The control classes may also be virtual representations of rerouting decisions; in that case these "virtual packets" will not create traffic overhead but will generate computational overhead at the nodes where decisions are taken. The probability that a user of class k is directed from router r to neighbour j by a control packet is $Q(r, k, j)$. Links will have only one predecessor and successor, while routers may have one or more successors that are links. Note also that some models may abstract the existence of links, and just represent the manner in which routers are connected without detailing the links. The equations of the network model are:

$$\Lambda_R(r,k) = \lambda(r,k) + \sum_{l\in\mathbf{L}} q(l,k)P(l,k,r)\mu_l(l),\ r\in\mathbf{R}$$
$$\Lambda_L(l,k) = \sum_{r\in\mathbf{R}}[P(r,k,l)q(r,k)\mu_r(r,k)$$
$$+\Lambda^-(r,(r,k))q(r,k)Q(r,k,l)],\ l\in\mathbf{L}$$
$$q(r,k) = \frac{\Lambda_R(r,k)}{\mu_r(r,k)+\Lambda^-(r,(r,k))},\ r\in\mathbf{R}$$
$$q(l,k) = \frac{\Lambda_L(l,k)}{\mu_l(l)},\ l\in\mathbf{L},$$

where $\Lambda_R(r,k),\ \Lambda_L(l,k),\ q(r,k),\ q(l,k)$ denote the total arrival rates to the routers and links, and the utilisation rate for the routers and links, for user traffic class k. The corresponding quantities for control traffic class (i,k) are given by:

$$\Lambda^-(j,(i,k)) = \lambda^-(j,(i,k)) + \sum_{l\in\mathbf{L}} p((i,k),l,j)c_L(l,(i,k))\mu_l,\ i,j\in\mathbf{R}$$
$$= \sum_{r\in\mathbf{R}} p((i,k),r,j)c_R(r,(i,k))\mu_r,\ i\in\mathbf{R}, j\in\mathbf{L},\ i\neq r$$
$$c_L(l,(i,k)) = \frac{\sum_{r\in\mathbf{R}} p((i,k),r,l)c_R(r,(i,k))\mu_r}{\mu_l}, l\in\mathbf{L}, i\in\mathbf{R}$$
$$c_R(r,(i,k)) = \frac{\lambda^-(r,(i,k)) + \sum_{l\in\mathbf{L}} p((i,k),l,r)c_L(l,(i,k))\mu_l}{\mu_r}$$

The steady-state probability that router r is busy is

$$B_R(r) = \sum_{k\in\mathbf{U}}[q(r,k) + \sum_{i\in\mathbf{R}} c_R(r,(i,k))] \tag{1}$$

and the steady-state probability that link l is busy is

$$B_L(l) = \sum_{k\in\mathbf{U}}[q(l,k) + \sum_{i\in\mathbf{R}} c_L(l,(i,k))] \tag{2}$$

The average network delay for the user traffic is then:

$$T_N = \frac{1}{\Lambda_T^+}[\sum_{r\in R}\frac{B_R(r)}{1-B_R(r)} + \sum_{l\in L}\frac{B_L(l)}{1-B_L(l)}] \tag{3}$$

where $\Lambda_T^+ = \sum_{k\in\mathbf{U}}\sum_{r\in\mathbf{R}}\lambda(r,k)$ is the total user traffic entering the network. The cost f to be minimised via judicious routing will contain a function of the probabilities that nodes and links are busy, and the power consumption of the network:

$$P_N = \sum_{r\in R} P(r) + \sum_{l\in L} P(l) \tag{4}$$

where router power consumption is represented by

$$P(r) = \alpha_r + g_R(B_R(r)) + c_r\sum_{k\in\mathbf{U}}\Lambda_R^-(r,(r,k)), r\in\mathbf{R} \tag{5}$$

and α_r is the router's static power consumption, $c_r > 0$ is a constant, $g_R(.)$ is an increasing function of the packet processing rate, c_r is a proportionality constant related to router processing for re-routing control, and power consumption in a link is:

$$P(l) = \beta_l + g_L(B_L(l)), l \in \mathbf{L} \tag{6}$$

where β_r is the static power consumption and $g_L(B_L(l))$ is an increasing function. The routing optimisation algorithm [21] minimises f which in general includes both network power consumption and average user packet delay:

$$\frac{Use\ Q(i,k,j)}{To\ Minimise}\ f = P_N + cT_N \tag{7}$$

where $c \geq 0$ is a constant the establishes the relative importance of delay with respect to power.

2.1 Improving upon Shortest-Path Routing

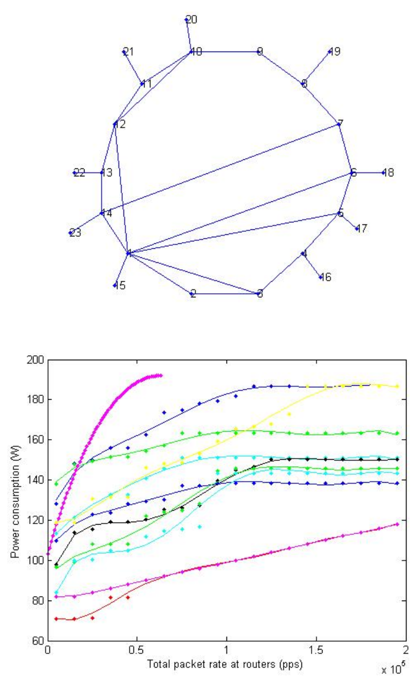

Fig. 1. Experimental network (top) and power profiles of 14 routers on the circle (bottom)

All comparisons are carried out based on the 23 node test-bed shown in Figure 1 described in [22, 20]. The measured power consumption of the fourteen nodes located on the circle are shown in the lower Figure 1. The service rates of the links are their 100 Mbps speeds. Also, virtual delays are added to service times so as to compensate for the short physical links that are used in the laboratory. The comparisons are carried out in the presence of flows travelling from source to destination with average traffic rates: Flow 1 (22,18) traffic rate 30kpps, Flow 2(23,19) traffic rate 10kpps, Flow 3 (21,17) traffic rate 20kpps. First, we apply the optimisation algorithm to the network started in a state where all flows follow the shortest path, and we focus on power ($c = 0$). We can select among seven alternative paths for each flow, and the optimisation yields a saving of 10 Watts, down from 1531 Watts, at the cost of an increase in average end-to-end delay of 3.3ms. Then we vary the input traffic of the 3 flows from 0.1 to 1.5 times their initial value, and the results in Figure 2 show a modest average power savings of 8.2 Watts while average packet delay increases. If we opt for both power and

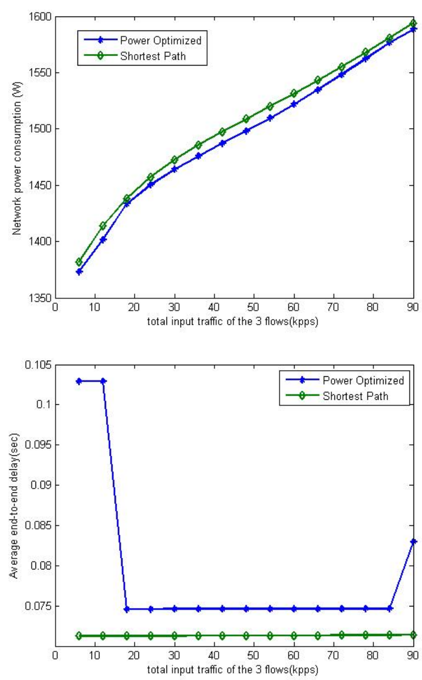

Fig. 2. Power consumption (top) and average end-to-end packet delay (bottom) against varying traffic load in Kpps (kilo-packets per second) for power-optimised versus shortest path routing

delay optimisation by adjusting c in (7) we can avoid the increase in average delay seen in Figure 3 and the average power savings is a modest but real 6.4 Watts. If we add another flow Flow 4 (20,16) at traffic rate 5kpps, and vary the

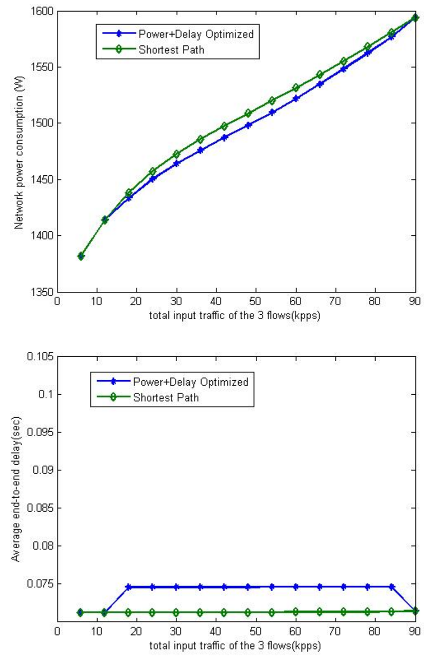

Fig. 3. Power consumption (above) and end-to-end delay with power and delay optimisation (below)

traffic of the four flows from 0.1 to 1.5 times their nominal values, the average power savings increase to 13.1 Watts as shown in Figure 4.

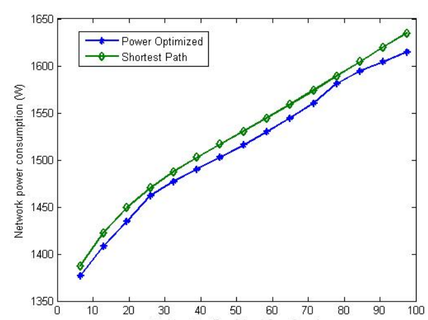

Fig. 4. Power consumption for proposed algorithm compared to shortest path with four flows and power optimisation

2.2 Improving upon EARP

We now use the gradient based otimisation scheme to improve upon the on-line adaptive power and QoS protocol EARP [22]. EARP uses CPN to search for the paths that minimise a mixed QoS and power consumption criterion. The log files of observed paths for the three flows as they are generated by EARP are then used to initiate the optimisation algorithm. For every new path observed by EARP we run the gradient algorithm, and the outcome is shown in Figure 5 where we observe a significant power saving with respect to EARP (dashed lines represent the average values). In order to limit out of order packet arrival and

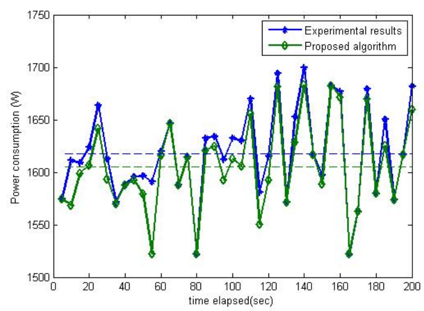

Fig. 5. Power consumption for the gradient algorithm (green) compared to power-based EARP [22] in blue

jitter, we can change paths only when the path modification improves on the previous power consumption; we then have the greater power savings of Figure 6.

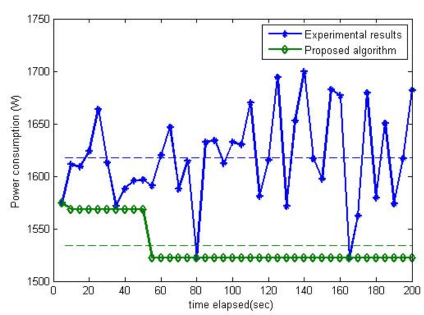

Fig. 6. Power consumption for proposed algorithm with memory

3 Conclusions

From the preceding discussion we see that the gradient based optimisation, started either with the network state set with the standard shortest-path algorithm for all of the flows in the network, or initially set using the self-ware EARP routing approach that adaptively reduces energy consumption, always brings some level of savings even if they are modest.

The impact on CO2 emissions will actually depend on the source of electrical energy, whether it is nuclear, or fossil, or renewable such as wind or electrovoltaic. However any energy savings will reflect at least at the same level in savings in the CO2 imprint of networks. We say “at least” because lower energy consumption would provide a better chance to run network nodes and link drivers for longer periods based on using renewable energy sources and reserve batteries when these sources are inactive. We note that if the experiments we have reported were scaled to a test-bed with high speed routers and large volumes of traffic, savings in energy costs and CO2 imprint can be substantial over long periods of time.

In previous work [21] we have shown that the algorithm we use in this paper is of time complexity $O(N^3)$, which would make it impractical for a large network. But in effect the algorithm can be simplified considerably because the gradient optimisation would in practice be only carried out for a limited number of nodes and a limited number of paths. For instance when we start with a shortest path, one will not look at all possible paths but rather work with other shortest paths or with paths which are at most one or two hops longer. Another simplification resides in the matrix inversion that leads to the $O(N^3)$ complexity. Rather than do a full matrix inversion, in most cases it may be sufficient to take the first two terms of the expansion of a matrix inversion $(I - W)^{-1} \approx (I + W + W^2)$ which can be faster than the matrix inversion. Also, any practical optimisation would be done in stages, working with successively smaller networks, or it may also be done hierarchically with a set of sub-network representations. We think that this part of the work can lead to many fruitful research options to study simplified algorithms and the impact that they will have on energy savings and QoS in the practical network.

Acknowledgment

The authors are grateful the research support of the Fit4Green EU FP7 Project co-funded under ICT Theme FP7-ICT-2009-4.

References

1. R. Rajaraman. Topology control and routing in ad hoc networks: A survey. *SIGACT News*, 33(2):60–73, 2002.

2. X. Jia, D. Li, and D. Du. QoS topology control in ad hoc wireless networks. In *INFOCOM 2004. 23rd Annual Joint Conf. IEEE Computer and Comms. Societies*, volume 2, pages 1264–1272, March 2004.
3. P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, 37(2):164–194, 2005.
4. A. Boukerche, X. Cheng, and J. Linus. A performance evaluation of a novel energy-aware data-centric routing algorithm in wireless sensor networks. *Wireless Networks*, 11(5):619–635, 2005.
5. E. Gelenbe and R. Lent. Power-aware ad hoc cognitive packet networks. *Ad Hoc Networks*, 2(3):205–216, 2004.
6. E. Gelenbe, E. Seref, and Z. Xu. Towards networks with intelligent packets. In *IEEE ICTAI 1999, IEEE Conference on Tools for Artificial Intelligence*, Chicago, IL, November 1999. IEEE.
7. E. Gelenbe. Steps towards self-aware networks. *Communications of the ACM*, 52:66–75, 2 2009.
8. X. Wang, A. V. Vasilakos, M. Chen, Y. Liu, and T. T. Kwon. A survey of green mobile networks: Opportunities and challenges. *ACM/Springer Mobile Networks and Applications*, doi:10.1007/s11036-011-0316-4, 2011.
9. M. Gupta and S. Singh. Greening of the Internet. *Computer Communication Review*, 33(4):19–26, 2003.
10. M. Gupta and S. Singh. Dynamic ethernet link shutdown for energy conservation on ethernet links. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 6156 –6161, June 2007.
11. J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power awareness in network design and routing. In *INFOCOM 2008. IEEE*, pages 457–465, April 2008.
12. N. Vasic and D. Kostic. Energy-aware traffic engineering. Technical report, EPFL, 2008.
13. S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall. Reducing network energy consumption via sleeping and rate-adaptation. In *NSDI'08: Proc. 5th Symp. on Networked Systems Design and Implementation*, pages 323–336, Berkeley, CA, USA, 2008. USENIX Association.
14. L. Chiaraviglio, M. Mellia, and F. Neri. Reducing power consumption in backbone networks. In *ICC '09. IEEE*, pages 1–6, June 2009.
15. L. Chiaraviglio, M. Mellia, and F. Neri. Energy-aware backbone networks: A case study. In *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, pages 1–5, June 2009.
16. L. Chiaraviglio, D. Ciullo, E. Leonardi, and M. Mellia. How much can the Internet be greened? In *GLOBECOM Workshops, IEEE*, pages 1–6, Dec. 2009.
17. E. Gelenbe and S. Silvestri. Reducing power consumption in wired networks. In *ISCIS'09, Computer and Information Sciences, 24th International Symposium on*, pages 292–297, METU, 14-16 September 2009. IEEE Digital Library.
18. E. Gelenbe. Cognitive packet network (CPN). U.S. Patent 6,804,20, October 11 2004. Energy-efficient cloud computing. *The Computer Journal*, 53(7):1045–1051, September 2010.
19. A. Berl, E. Gelenbe, M. D. Girolamo, G. Giuliani, H. D. Meer, M. Q. Dang, and K. Pentikousis. Energy-efficient cloud computing. *The Computer Journal*, 53(7):1045–1051, September 2010.
20. R. Lent. A sensor network to profile the electrical power consumption of computer networks. In *GLOBECOM 2010 Workshop on Green Computing and Communications*. IEEE, December 2010.

21. E. Gelenbe and C. Morfopoulou. A framework for energy-aware routing in packet networks. *The Computer Journal*, 54(6):850–859, 2011.
22. E. Gelenbe and T. Mahmoodi. Energy-aware routing in the cognitive packet network. In *International Conf. on Smart Grids, Green Communications and IT Energy-Aware Technologies - Energy 2011*, May 2011.
23. E. Gelenbe. A unified approach to the evaluation of a class of replacement algorithms. *IEEE Transactions on Computers*, 22(6):611–618, June 1973.
24. E. Gelenbe. G-networks with signals and batch removal. *Probability in the Engineering and Informational Sciences*, 7:335–342, 1993.
25. E. Gelenbe. G-networks with instantaneous customer movement. *Journal of Applied Probability*, 30(3):742–748, 1993.
26. E. Gelenbe and A. Labed. G-networks with multiple classes of signals and positive customers. *European Journal of Operations Research*, 108(2):293–305, 1998.
27. E. Gelenbe and A. Stafylopatis. Global behaviour of homogenous random systems. *Applied Mathmetical Modelling*, 15(10):534–541, October 1991.