Energy-Aware Routing in the Cognitive Packet Network

Toktam Mahmoodi

Intelligent Systems & Networks Group
Department of Electrical & Electronic Engineering
Imperial College, SW7 2BT London UK
t.mahmoodi@imperial.ac.uk

Abstract

An energy aware routing protocol (EARP) is proposed to minimise a performance metric that combines the total consumed power in the network and the QoS that is specified for the flows. The algorithm uses source routing based on the functionalities provided by the Cognitive Packet Network (CPN), running autonomously at each input node to the network based on smart packets which gather relevant information throughout the network using reinforcement learning at each of the intermediate nodes. Measurements on an experimental test-bed that uses EARP are presented and they indicate that it offers a reduction in power consumption, as compared to a purely QoS driven approach, and also respects the requested QoS level.

Keywords: Energy Efficiency; Routing Protocol; Cognitive Packet Network

1. Introduction

Energy efficient protocols have been extensively studied for wireless networks, because energy savings for battery powered nodes is crucial [1, 2]. However research on energy consumption is relatively new in wired networks even though the amount consumed on a day to day basis is a significant fraction of the total energy used for ICT systems [3]. Surprisingly the total energy consumption for ICT and for air travel are comparable. Since wired networks form the backbone of all of the world's ICT, the energy consumption in this area is bound to rise unless serious efforts are made to achieve significant savings in wired networks and computer systems. Thus recent research on wired energy aware network management includes [4, 5, 6, 7, 8, 9].

The most thorough measurement studies that have been performed in [10] which quantify the energy consumed by many network devices, ranging from the core switches to wireless access points, and including different vendors. These measurements are carried out under various traffic and network configurations, together with an index associated with each network device so that the proportionality of power consumption to the device's traffic load can be evaluated. As investigated by [10], the ratio of the actual power consumed by a networking device on average to its maximum power consumption, varies widely across different device families. The impact of the hardware processing rate and traffic load on power consumption is also examined in [4]. Moreover, research work in [11] introduces a generic model for router power consumption.

However because there does not seem to be a single unified model that captures the power characteristics of a wide class of network devices, in the experiments we conducted in this paper, we use offline power measurements that have been conducted on our own experimental testbed's nodes, and which have been previously reported in [12, 13]. In particular, we will rely on the measurements reported in Figure 1 for a single core router to relate router traffic rates in packets per second to the power consumed by each of our routers.

In this paper we propose an energy aware routing protocol (EARP) that not only attempts to minimise the total consumed power in the network but also respects the requested QoS by each incoming flow [14]. EARP relies on the underlying Cognitive Packet Network (CPN) [15] for the information it requires, and uses it to minimise power consumption. CPN's smart packets are used to gather information about the power usage at the nodes, and EARP is run in a fully distributed manner using CPN's source routing scheme that is modified to include power consumption as a decision criterion.

The remainder of this paper is organised as follows. We first give a brief overview of CPN and its existing routing protocol. Section 2 elaborates the proposed energy-aware routing protocol. The implementation is summarised and then we detail some performance results in Section 3. Conclusions and further work are discussed in Section 4.

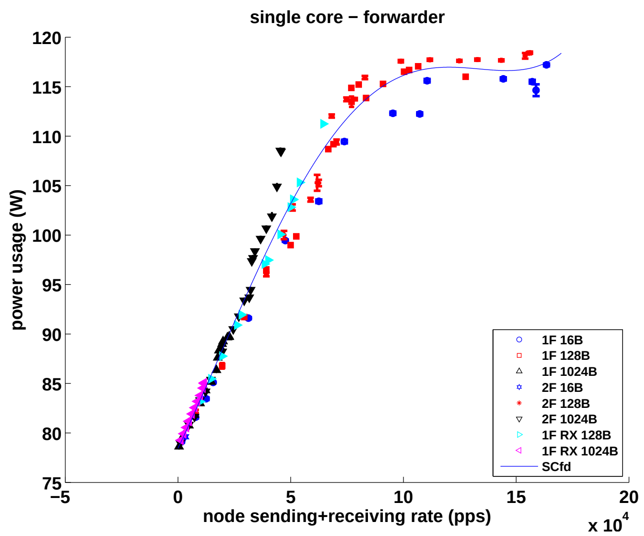


Figure 1: Power consumption as a function of packet rate for a single core router [12]

1.1. *Background*

The Cognitive Packet Network (CPN) is an experimental protocol that allows a network with an arbitrary topology to observe its state in a distributed manner and exploit the data being gathered to improve different QoS metrics [16]. The CPN routing algorithm runs autonomously at each node using Reinforcement Learning with a recurrent Random Neural Network [17, 18, 19], and measurement results for this protocol are summarised in [20]. CPN makes use of three types of packets: smart packets (SP) for discovery, source routed dumb packets (DP) to carry payload, and acknowledgements (ACK) to bring back information that has been discovered by SPs. Conventional IP packets can also tunnel through CPN, so as to seamlessly operate mixed IP and CPN networks. SPs are constantly generated by each of the source users of CPN as a fraction of the total number of DPs that are sent forward, in order to seek paths to the destination that minimise the desired QoS, and also to update the QoS information about different paths used by the source and hence allow it to make informed decisions. Since DPs are source routed, the choice of the path that is used to convey the DPs to the destination, is made at the source on the basis of the information it receives from ACKs sent back by the successful SPs. In previous work CPN has been proposed as a means to optimise energy consumption [5, 6], and this paper is a continuation of this previous research. However here we will use a decision scheme that attempts to minimise the power consumed provided the overall end-to-end packet delay remains below a predetermined value.

2. Energy and QoS Aware Routing

At each node i, let us denote by T_i, the traffic this node carries in packets/sec (pps). Assuming that a flow l carries traffic of rate t_l pps, then T_i can be computed as

$$T_i = \sum_{l \in F(i)} t_l \tag{1}$$

where $F(i)$ denotes the set of flows that use node i. Let $p_i(T)$ and $Q_i(T)$ the power consumption and QoS of node i when the traffic it carries is T, so that adding a new flow k to node i will result in a change of power consumption and QoS at that node.

Let $p_i(x)$ be the instantaneous power consumption at node i when it carries x packets per seconds, in watts, will include all aspects of packet processing, including storing packets, routing them, and forwarding them through appropriate link drivers. It is of course possible to detail these different elements of power expenditure. In modern routers $p_i(x)$ will increase with x, but because of the increasing use of multi-cores in processing elements, the increase may have a step-like behaviour.

Let us denote the *Power Cost* associated with the $k-th$ flow at node i by $M_i^k(t_k, T_i)$, and define it as a combination of the flow's own power consumption, and of the impact it has on other flows which are using the node:

$$M_i^k(t_k, T_i) = at_k \frac{p_i(T_i + t_k)}{T_i + t_k} + bT_i[\frac{p_i(T_i + t_k)}{T_i + t_k} - \frac{p_i(T_i)}{T_i}] \quad (2)$$

where $a,\ b \geq 0$. Here the first term is the power (watts) total used by the $k-th$ flow, multiplied by some constant a. The second term represents the increase in wattage for the other flows, multiplied by some constant b. Note that if $a = b = 1$ both of these elements have an equal weight, while if $b = 0$ then we are ignoring the effect on the other flows that are using the node. However this metric assumes that the $k - th$ flow is concerned with a form of payment of "wattage per packet" that may be paid. In fact, a flow may also be concerned just with the total wattage itself, in which case a more appropriate metric may be:

$$m_i^k(t_k, T_i) = cp_i(T_i + t_k) + d[p_i(T_i + t_k) - p_i(T_i)] \quad (3)$$

for $c,\ d \geq 0$. Note that for any quality of service function $Q_i^k(t_k, T_i)$ we can adopt similar forms as (2) and (3), but generally a simplified version of the latter may be adopted for quantities such as node delay and packet loss.

The power related cost functions for the k-th traffic flow of rate t_k on a path $\pi(i)$ originating at node i is written as:

$$M_{\pi(i)}^k(t_k, \overline{T_{\pi(i)}}) = \sum_{n \in \pi(i)} M_n^k(t_k, T_n), \quad (4)$$

or we can choose the simpler form:

$$m_{\pi(i)}^k(t_k, \overline{T_{\pi(i)}}) = \sum_{n \in \pi(i)} m_n^k(t_k, T_n), \quad (5)$$

Similarly, we would have the QoS criterion, such as loss, delay or some other metric:

$$Q_{\pi(i)}^k(t_k, \overline{T_{\pi(i)}}) = \sum_{n \in \pi(i)} Q_n^k(t_k, T_n) \quad (6)$$

where $\overline{T_{\pi(i)}} = (T_{n_1},\ \ldots\ , T_{n_{|\pi(i)|}})$ where $n_1 = i$, and the n_j, with $1 \leq j \leq |\pi(i)|$ are the successive nodes of path $\pi(i)$.

The main drawbacks of using the metric $M_{\pi(i)}^k(t_k)$ are twofold. (a) Because of the factor $t_k/(t_k + T_i)$, the first term may be quite small, and the second term may also be small because we compute a difference in energy consumption. Small values compounded with the effect of inevitable statistical fluctuations in measurements make this metric unattractive. (b) The need to measure three quantities at each node plus the traffic rate t_k can lead to excessive overhead and measurement delays. Thus it appears more attractive and much simpler to use $m_{\pi(i)}^k(t_k, \overline{T_{\pi(i)}})$ as the energy criterion to be optimised.

2.1. Reinforcement Learning in EARP

In CPN, each router stores a specific Random Neural Network (RNN) for each flow that is active at that node. Each RNN has as many neurons as there are outgoing links in the node. The arrival of a smart packet (SP) will trigger the interrogation of the RNN to determine the next hop for the SP; this is done by computing the current state of the RNN and selecting the output port of the node that corresponds to the neuron of the RNN which is the most excited. On the other hand, the arrival of an acknowledgement packet (ACK) back from the destination of that flow, will trigger the execution of the reinforcement learning (RL) process [18]. Since EARP is expected to minimise the overall cost of power while satisfying the requested QoS, the goal G_i to be optimised will combine the power consumption with the QoS constraint. All quantities of interest for some flow k will relate to the forward path from any node i to the destination node of that flow. Thus the goal will take the form:

$$\begin{aligned} G_i &= m^k_{\pi(i)}(t_k, \overline{T_{\pi(i)}}) \\ &+\beta 1[Q^k_{\pi(i)}(t_k, \overline{T_{\pi(i)}}) - Q^k_o > 0](Q^k_{\pi(i)}(t_k, \overline{T_{\pi(i)}}))^\nu \end{aligned} \tag{7}$$

where:

- $m^k_{\pi(i)}(t_k, \overline{T_{\pi(i)}})$ is the total power cost function on the path going from the $i-th$ node to the destination of flow k, with the corresponding traffic loads on each of the intermediate nodes,
- $1[X]$ is the function that takes the value 0 if X is true, and takes the value 1 if X is false.
- $\nu \geq 1$, and $\beta > 0$ is a constant meant to match the delay units with respect to power, while Q^k_o is the QoS value that *should not be exceeded* for flow k, e.g. the maximum tolerated path delay,
- $Q^k_{\pi(i)}(t_k, \overline{T_{\pi(i)}}))$ is the total QoS value measured from this node to the destination by the SPs.

CPN uses the reward function $R = G^{-1}$ as follows. If the R_θ are the successive measured values of the reward function R at some node, then the RNN weights are updated based on the threshold Θ_θ, which captures a historical sliding window average of the reward:

$$\Theta_\theta = \alpha\Theta_{\theta-1} + (1-\alpha)R_\theta, \tag{8}$$

where he constant value $0 \leq \alpha \leq 1$ tunes the responsiveness of the algorithm, i.e. $\alpha = 0.8$ represents an “average sliding window” of the five past values of R_θ. Weights are increased or reduced or based on the difference between the current reward R_θ and the previous threshold $\Theta_{\theta-1}$; if R_θ is larger than $\Theta_{\theta-1}$ then this results in a significant increase in the excitatory weights from all neurons to that neuron that is ignited the previous output link, with a slight increase in the inhibitory weights leading to other neurons. Otherwise, if R_θ is

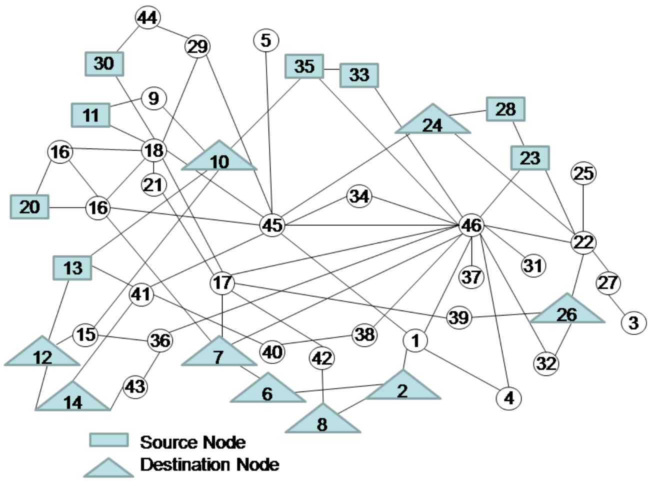


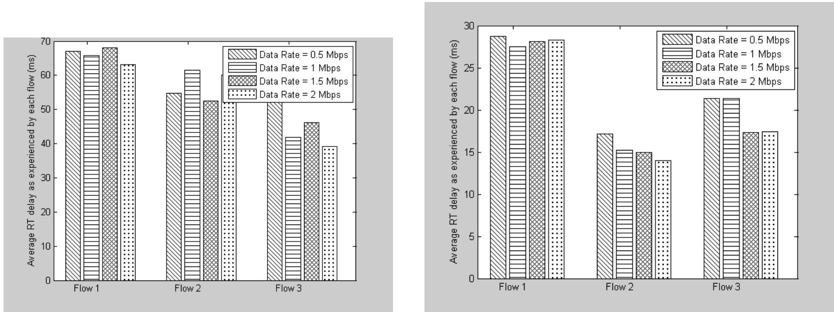
Figure 2: Network topology.

smaller than $\Theta_{\theta-1}$, all excitatory weights leading to all neurons are moderately increased, except for the previous winner, and the inhibitory weights leading to the previous winning neuron are significantly increased, in order to "punish" it for not being successful. $\Theta_{\theta-1}$ is first computed, and then the network weights are updated as follows. In the following expression that, the neurons i, j, n correspond to output links of the node where the update is being conducted, and none of these output links can be identical to the link from which the connection (for which the updates are being carried out) has entered this particular node. Thus, if the node has N links, one of them is excluded because the input link for the connection cannot be re-used as the output link (i.e. packets cannot be sent back along the link through which they entered), j denotes the output link that was most recently used by the connection, and hence there are $N-2$ alternate output links to be considered:

$$\begin{aligned}
\Theta_{\theta-1} \le R_\theta &: \begin{cases} w^+(i,j) & \leftarrow \quad w^+(i,j) + R_\theta, \\ w^-(i,n) & \leftarrow \quad w^-(i,n) + \frac{R_\theta}{N-2}, \ \forall n \neq j, \end{cases} \\
\Theta_{\theta-1} > R_\theta &: \begin{cases} w^-(i,j) & \leftarrow \quad w^-(i,j) + R_\theta, \\ w^+(i,n) & \leftarrow \quad w^+(i,n) + \frac{R_\theta}{N-2}, \ \forall n \neq j. \end{cases}
\end{aligned} \tag{9}$$

3. Experiments

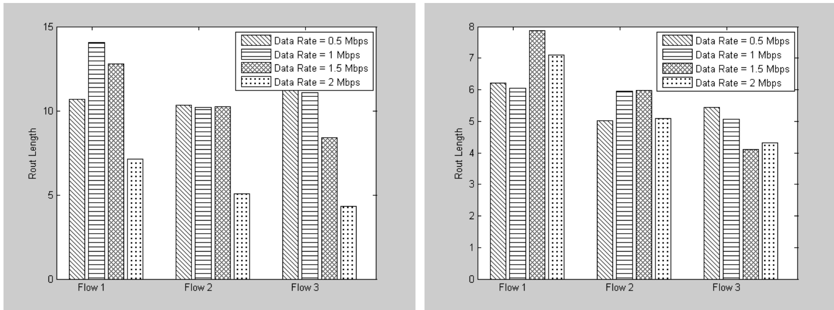
Our experimental testbed consists of 46 nodes, which are Pentium IV-machines with up to fifteen Ethernet interfaces running Linux Kernel 2.6.15. These nodes are connected using a topology depicted in Figure 2, with full-duplex links at



(a) Routing based on energy with delay constraint

(b) Routing based on delay

Figure 3: Average round trip delay for the three flows with different traffic levels



(a) Routing based on energy with delay constraint

(b) Routing based on delay.

Figure 4: Average length of the end-to-end path taken by the three active flows.

10 Mbps Mega-bits-per-second). The topology we have selected resembles that of the Swiss Education and Research Network, and artificial delays are used to replicate the link-level delays of the real network. We further assume that all the nodes have the same power consumption characteristic as a function of traffic rate as in Figure 1 that was measured for the single core routers that are used in our network test-bed, and the function p_i, that relates traffic rate to power consumption in node i is approximated by a piecewise linear function. In the EARP, we have chosen a maximum value of acceptable end-to-end delay of $Q_o^k = 80$ms. The constants $\nu = 8$ and $\beta = 1$ so that the second term in (8) tends to become very large when the end-to-end delay approaches 80ms, so that this delay is never exceeded. All delays are expressed and measured in milliseconds.

3.1. Experiments with three source-destination pairs and no background traffic

We first chose three source and destination nodes, as indicated in Figure 2, and set up three flows: from node 19 to 3, from node 30 to 2 and from node 33 to 14. The three flows have the same data rate, which was varied with four values 0.5, 1, 1.5 and 2 Mbps. There was no other traffic in the network in this first experiment.

All the experiments were based on UDP traffic, and packet size was fixed at 1024 bytes. Each experiment ran for 600 secs, and measurements were collected from each node every five seconds. Additional background traffic at a rate of 200 kbps was also set up to run alternately every other 100 seconds over all the links between nodes 45-24, 35-10, and 46-38 in both directions, so as to create a time varying power load as well as a time-varying power consumption pattern in the network.

With this configuration, we compare the performance of EARP with that of the CPN protocol which aims to minimise end-to-end delay. We thus measured the power consumption in all the nodes of the network and the round trip delay experienced by the active flows.

All three flows were first run at a data rate of 0.5 Mbps, which was then increased in steps of 0.5 Mbps for each successive round of the experiment simultaneously for all three flows, up to the maximum value of 2 Mbs. Experiments were conducted separately with the EARP, and also with conventional CPN that was using delay only as the QoS goal.

On the other hand, as would be expected, EARP results in higher end-to-end delays as shown in Figure 3, mainly due to the longer paths taken by EARP to avoid nodes that carry more traffic and hence which consume more power per packet. To detail this point, the average length of the end-to-end paths used by each of the two schemes are plotted in Figure 4. It can be seen that the routes selected by CPN with delay minimisation are on average 40% shorter than those selected by EARP. Furthermore, Figure 3 also reveals that although delay may increase with EARP, each flows' round trip delay remains within the prescribed limit. This suggests that we could also modify EARP to include other QoS bounds, such as loss, a combination of loss and delay, and jitter.

3.2. Network-wide Energy Savings

We repeated the previous experiments with three flows, but also additional background traffic was added running throughout all the network links, in each direction at a constant rate of 200 kbps.

The resulting measured total power consumption over all nodes is shown in Figure 5 for routing based on energy, on delay and on shortest path, and for the different traffic rates on each connection. As expected, the routing based on energy (EARP) results in the lowest energy consumption, while routing based on shortest path which is insensitive to load conditions results in the highest energy consumption because energy consumption itself depends on load. Also as expected, we see that delay based routing provides a compromise since at low traffic levels it results in comparable energy consumption to EARP, while at high traffic loads it does not do as well as EARP.

Figure 6 compares the *average route length* in number of hops, averaged over all traffic levels for the three connections, for EARP on the one hand, and CPN that is based on using the number of hops as the QoS criterion. Note that the

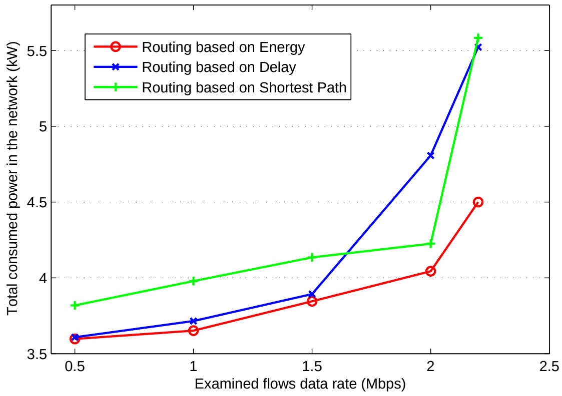


Figure 5: Experiment with three flows: total power consumption in the network vs. traffic rate all connections.

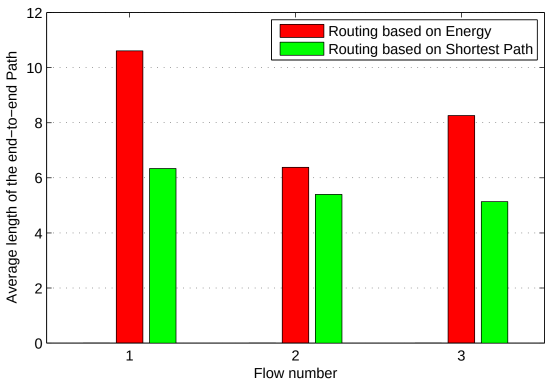


Figure 6: Experiment with three flows: average length of the paths taken by the connections.

averages that are taken ar not "per packet", but rather per experiment. We see that EARP can result in significantly longer path lengths being taken, so that short path lengths obviously will not in general lead to lower overall energy consumption per path.

Figure 7 on the other hand examines the average round-trip delay experienced by packets, where again the average is taken over all the different traffic intensities and experiments for all of the three connections, where we compare EARP with the CPN protocol that attempts to minimise delay. We see that EARP can lead to significantly higher delays in its attempt to minimise energy consumption.

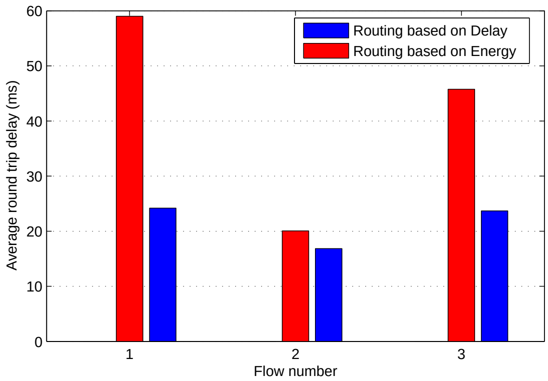


Figure 7: Experiments with three flows: average round trip packet delay for the three connections.

3.3. Experiments with nine connections

In a second set of experiments, we take nine (rather than three) source-destination pairs. Three flows are initially activated in the network, and then during the experiment three additional connections are launched, followed by three more. The three flows activated at the start of the experiment go from node 23 to 12, 30 to 14 and 33 to 2. Another three flows are initiated 100 secs later (from node 20 to 26, from node 28 to 10 and from node 35 to 7), and the finally three flows are activated 200 secs after the start from node 11 to 8, node 13 to 24 and node 29 to 6. Each flow's lifetime is 400 secs, and the total experiment lasts for 600 secs. All nine flows are first run at a data rate of 0.5 Mbps, and the data rate is then increased to 1, 1.5, 2, and 2.2 Mbps in successive steps. In addition we have 200 kbs of background traffic being conveyed in both directions over every link in the network.

The instantaneous power consumed by the network for flow rates of 1 Mbps and 2 Mbps is shown in Figure 8 during the first 400 secs of the experiment while all the nine flows are still active, as well as the step increases in power consumption at 100sec and 200 secs, when three new flows are initiated each time. We observe the saving in power consumption when the EARP is used for both flows' rates 1 Mbps and 2 Mbps, as compared to shortest and delay based routing.

On the other hand, in Figure 9 we observe average values over the whole experiment (rather than instantaneous values) of power for five different data rates of the nine flows, and we see that using EARP always results in savings in power consumption. The irregularity in the curve for the power consumed using shortest path routing with CPN just indicates that CPN shortest path routing can actually use different paths, and these different paths can result in different levels of energy consumption that do not necessarily result in an overall increase when traffic rates increase on each flow.

In Figure 10 we measure the flows' end-to-end delay obtained using EARP,

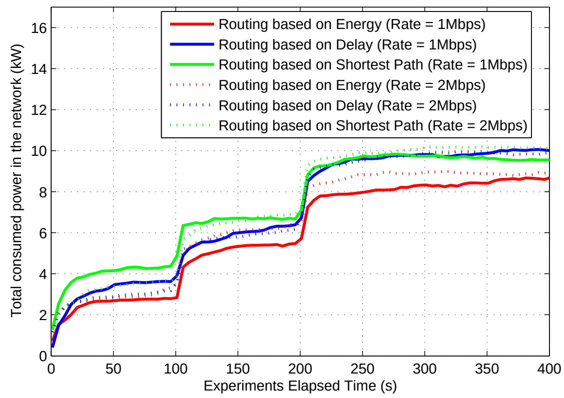


Figure 8: Scenario two: Total power consumption in the network Vs. the experiment's elapsed time.

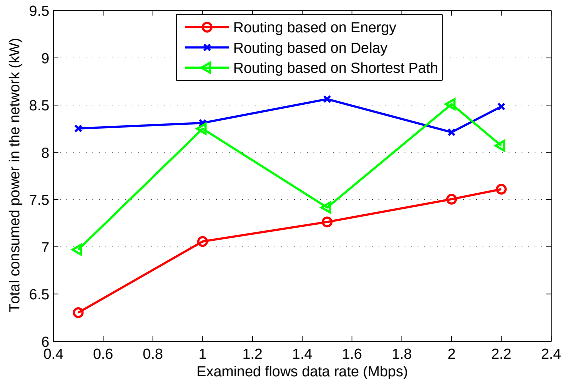


Figure 9: Scenario two: Total power consumption in the network vs. traffic rate of the examined flows

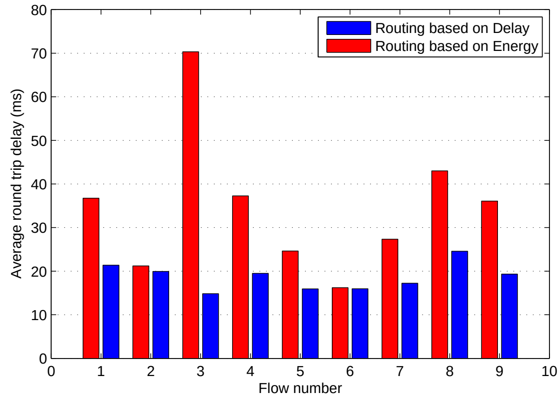


Figure 10: Scenario two: Average round trip delay for each of the nine active flows in their lifetime.

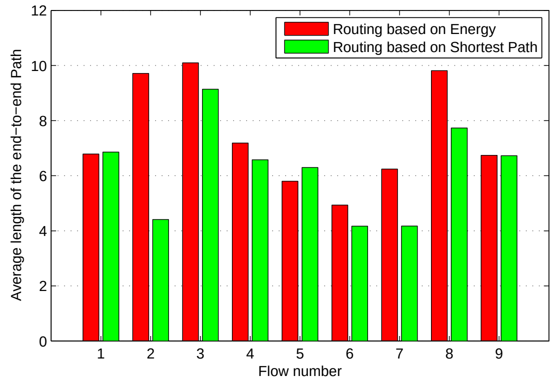


Figure 11: Scenario two: Average length of the end-to-end path taken by each of the nine active flows.

and compare it with the delay resulting from using CPN with delay minimisation. The average delay for each connection is presented, where the average is computed for each flow over the five data sets that we have tested. We observe that CPN with delay minimisation results on average in a 45% reduction of latency as compared to EARP. On the other hand, Figure 11 shows the average path length used by the EARP in number of hops for each connection with respect to the CPN protocol that uses the minimisation for the number of hops (rather than the minimisation of delay); we see that the difference is perceptible (0.6 hops on average) but not very significant.

4. Conclusions

This paper introduces a novel energy-aware routing protocol (EARP) that is based on the autonomic network routing protocol CPN that was described in several other papers. EARP attempts to minimise the total power consumption of each flow in a packet network, while trying to keep the "damage" to the delay experienced by packets principal QoS metric to a value which is below an acceptable upper bound.

We have implemented EARP on a large network test-bed with many different traffic conditions. We have conducted both averaged measurements for long periods, and also observed the network as the number of connections varies with time.

We have thus experimentally compared EARP's performance to a version of CPN that uses either just delay as the desired QoS metric to be minimised, or that attempts to minimise the number of hops used by the flows. We observe that EARP, which explicitly addresses power consumption does indeed provide the desired results, and that aiming at delay or hop count minimisation will result in significantly larger energy consumption levels. This comparison has shown that EARP achieves overall reduction of power consumption on average throughout the network, but also that the QoS metric of interest (which was delay in our experiments) increases, though it remains below a maximum acceptable level.

Our measurements have estimated power consumption via traffic measurements using the directly measured relationship between power consumption and traffic rate in routers. We expect that we can improve the accuracy of our power estimates in the future by using specific sensors which directly access the power consumption of routers and of other network equipment. In future work we also plan to address more complex QoS metrics which incorporate factors such as loss, jitter as well as delay.

The main drawback of EARP may be that even when traffic rates are such that energy consumption is low, it can unnecessarily degrade QoS. Therefore in future work we will consider changes to EARP that would only launch the algorithm in ranges of traffic and energy consumption where significant improvement of energy consumption is obtained without significant negative effects on QoS.

Acknowledgements

The author would like to thank Prof. Erol Gelenbe for his great support and major contribution to this work. The support and technical motivation that has been provided to this research by the EU FP7 Fit4Green Project is also greatly recognised.

References

[1] S. Cui, R. Madan, A. Goldsmith, S. Lall, Joint routing, MAC, and Link Layer Optimization in Sensor Networks with Energy Constraints, in: IEEE International Conference on Communications (ICC '05), 725–729, doi: 10.1109/icc.2005.1494448, 2005.

[2] E. Gelenbe, R. Lent, Power-aware Ad Hoc Cognitive Packet Networks, Ad Hoc Networks 2 (3) (2004) 205–216, doi:10.1016/j.adhoc.2004.03.009.

[3] A. Berl, E. Gelenbe, M. D. Girolamo, G. Giuliani, H. de Meer, M. Q. Dang, K. Pentikousis, Energy-Efficient Cloud Computing, The Computer Journal 53 (7) (2010) 1045–1051, doi:10.1093/comjnl/bxp080.

[4] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, D. Wetherall, Reducing Network Energy Consumption via Sleeping and Rate-Adaptation, in: the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI '08), 323–336, 2008.

[5] E. Gelenbe, S. Silvestri, Reducing power consumption in wired networks, in: the 24th International Symposium on Computer and Information Sciences (ISCIS '09), IEEE, 292–297, doi:10.1109/iscis.2009.5291829, 2009.

[6] E. Gelenbe, S. Silvestri, Optimisation of Power Consumption in Wired Packet Networks, in: the 6th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE '09), vol. 22 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST)*, Springer, 717–729, doi:10.1007/978-3-642-10625-5, 2009.

[7] C. Panarello, A. Lombardo, G. Schembra, L. Chiaraviglio, M. Mellia, Energy Saving and Network Performance: a Trade-off Approach, in: the 1st International Conference on Energy-Efficient Computing and Networking (e-Energy '10), ACM, 41–50, doi: http://doi.acm.org/10.1145/1791314.1791321, 2010.

[8] E. Gelenbe, C. Morfopoulou, A Framework for Energy Aware Routing in Packet Networks, Accepted for publication in The Computer Journal doi: 10.1093/comjnl/bxq092, first published online: December 15, 2010.

[9] E. Gelenbe, C. Morfopoulou, Routing and G-Networks to Optimise Energy and Quality of Service in Packet Networks, in: the 1st International ICST conference on E-Energy (E-Energy '10), vol. 54 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST)*, Springer, 163–173, 2011.

[10] P. Mahadevan, P. Sharma, S. Banerjee, P. Ranganathan, A Power Benchmarking Framework for Network Devices, in: the 8th International IFIP-TC 6 Networking Conference (Networking '09), Springer-Verlag, 795–808, doi:10.1007/978-3-642-01399-7, 2009.

[11] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, S. Wright, Power Awareness in Network Design and Routing, in: the 27th Conference on Computer Communications (INFOCOM '08), IEEE, 457–465, doi: 10.1109/infocom.2008.93, 2008.

[12] R. Lent, Simulating the Power Consumption of Computer Networks, in: the 15th IEEE International Workshop on Computer Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD '10), 96–100, doi:10.1109/camad.2010.5686955, 2010.

[13] R. Lent, A sensor network to profile the electrical power consumption of computer networks, in: IEEE Global Communications Conference (GLOBECOM '10), Workshop on Green Computing and Communications, 2010.

[14] E. Gelenbe, T. Mahmoodi, Energy-Aware Routing in the Cognitive Packet Network, in: International Conference on Smart Grids, Green Communications, and IT Energy-aware Technologies (Energy '11), 2011.

[15] E. Gelenbe, Cognitive Packet Network, U.S. Patent 6,804,201 (October 11, 2004) .

[16] E. Gelenbe, R. Lent, A. Nunez, Self-Aware Networks and QoS, Proceedings of the IEEE 92 (9) (2004) 1478–1489, doi:10.1109/jproc.2004.832952.

[17] E. Gelenbe, Learning in the Recurrent Random Neural Network, Neural Computation 5 (1) (1993) 154–164, doi:10.1162/neco.1993.5.1.154.

[18] E. Gelenbe, E. Şeref, Z. Xu, Simulation with Learning Agents, Proceedings of the IEEE 89 (2) (2001) 148–157, doi:10.1109/5.910851.

[19] U. Halici, Reinforcement Learning with Internal Expectation for the Random Neural Network, European Journal of Operational Research 126 (2) (2000) 288–307, doi:10.1016/S0377-2217(99)00479-8.

[20] E. Gelenbe, Steps Toward Self-Aware Networks, Communications of the ACM 52 (7) (2009) 66–75, doi: http://doi.acm.org/10.1145/1538788.1538809.