

Article

Investigating the Tradeoffs between Power Consumption and Quality of Service in a Backbone Network

Georgia Sakellari ^{1,*}, Christina Morfopoulou ² and Erol Gelenbe ²

¹ Computer Science Field, School of Architecture, Computing and Engineering (ACE), University of East London, London E16 2RD, UK

² ISN Group, Electrical & Electronic Engineering Department, Imperial College London, London SW7 2BT, UK; E-Mails: c.morfopoulou@imperial.ac.uk (C.M.); e.gelenbe@imperial.ac.uk (E.G.)

* Author to whom correspondence should be addressed; E-Mail: g.sakellari@uel.ac.uk; Tel.: +44-20-8223-7927.

Received: 8 February 2013; in revised form: 12 April 2013 / Accepted: 9 May 2013 /

Published: 24 May 2013

Abstract: Energy saving in networks has traditionally focussed on reducing battery consumption through smart wireless network design. Recently, researchers have turned their attention to the energy cost and carbon emissions of the backbone network that both fixed and mobile communications depend on, proposing primarily mechanisms that turn equipments OFF or put them into deep sleep. This is an effective way of saving energy, provided that the nodes can return to working condition quickly, but it introduces increased delays and packet losses that directly affect the quality of communication experienced by the users. Here we investigate the associated tradeoffs between power consumption and quality of service in backbone networks that employ deep sleep energy savings. We examine these tradeoffs by conducting experiments on a real PC-based network topology, where nodes are put into deep sleep at random times and intervals, resulting in a continuously changing network with reduced total power consumption. The average power consumption, the packet loss and the average delay of this network are examined with respect to the average value of the ON rate and the ON/OFF cycle of the nodes.

Keywords: power consumption; network performance; quality of service

1. Introduction

The carbon imprint of ICT technologies is estimated to be over 2% of the world total, similar to that of air travel [1]. Until recently, energy saving in networks focussed more on longer battery life of mobile devices and smart design of the wireless infrastructure [2–4], and more attention has been devoted to energy consumption of Cloud Computing [5,6]. Traditionally, the energy cost of the much more energy demanding backbone networks that support Internet communications, both fixed and mobile, has not been taken into account, but researchers have recently turned their attention to the energy cost and the carbon emissions of the backbone network [7,8].

In this paper we investigate the effect of deep sleep based power savings in the backbone network on the quality of service (QoS), and more specifically, the tradeoff between power consumption and network delay and packet loss. We propose a novel queuing mechanism for routers to store outgoing packets that would otherwise be lost if the next node on their route were sleeping.

The remainder of this paper is organised as follows: we start with a review of the various approaches in the use of deep sleep and related mechanisms for energy saving. We continue with a detailed description of our proposed neighbour-aware packet queuing mechanism, then we describe the network configuration for our experiments and their results on a real large scale testbed. We conclude with the summary of our findings and suggestions of future work.

2. Previous Work

Early work has studied the consequences on network quality of service, including delay, packet loss and jitter, of putting routers OFF and ON as a way to save energy and adapt energy consumption to actual network load [9,10]. Recent studies also indicate that energy can be saved by modifying routing policies, for example, by aggregating traffic along a few routes and switching OFF, putting in standby or putting to sleep some network nodes. One such approach is to use dynamic bandwidth allocation, as a way to allow some users to have increased bit rate when others do not require as much [2]. Another approach is dynamic topology optimization where the alterations of the load are exploited in order to select among all topologies that ensure connectivity and satisfy the traffic demands the one that has the lowest overall power consumption [2]. Another technique called proxying transfers the management of traffic to a dedicated proxy entity that stores all packets and responds to requests for connections, enabling other network nodes to sleep for longer periods [11]. In [12], the authors propose an enhancement to the OSPF routing protocol in which a subset of IP routers are selected to be switched OFF during periods of low traffic. In [13] a greener router is proposed by implementing congestion control and capacity scaling.

The recent survey [8] describes several techniques for achieving energy efficiency in networks. Early work suggesting energy savings in the Internet [14] proposed routing modifications so as to aggregate traffic along a few routes, leading to a modification of the network topology based on route adaptation and putting nodes and devices to sleep. More recent work illustrated the feasibility of using admission control to restrict the more energy demanding users [15]. The problem of energy aware routing is examined via an analytical approach in [16] where an optimization algorithm is built in order to minimise a composite energy and QoS cost function. In [17] an autonomic algorithm based on the CPN protocol [18] is proposed to minimise power consumption while meeting the requested end-to-end delay bounds.

In [19] two widely used routers are measured in terms of system power demand with different configurations of line cards and operating conditions. While traffic was shown to have some impact on the power consumption of a line card, it is measured to be only around 2%. A similar power consumption model [20] includes the effect of individual ports configured on each line card, suggesting that the impact of port utilization on power consumption is under 5% and identifying the challenge for device manufacturers to ensure that networking devices are energy proportional. These studies indicate that the power consumption of current networking equipment is largely independent of its utilization and thus there is significant energy waste when nodes are inactive or lightly loaded.

An energy-aware online technique is proposed in [21] to spread the load through multiple paths, based on assuming 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 is also examined in [22] based on the utilization and the link queuing delay. Moreover, a sleeping approach is proposed, where traffic is sent out in bursts at the edge routers enabling other line cards to sleep between successive bursts of traffic. In [23], the authors select the active links and routers to minimise the power consumption via simple heuristics that approximately solve a corresponding NP-hard problem, while in [24], they discuss a case study based on specific backbone networks, and they propose an estimate of the potential overall energy savings in the Internet ([25]).

In [26], the relationship between network robustness, performance and Internet power consumption is investigated based on data collected from Internet sources. The authors approximate a generalized router power consumption function based on those values and propose a modeling framework taking into consideration the tradeoffs between power and performance. In [27] the authors formulate green routing as a mixed integer optimization problem and consider several power models corresponding to different technologies and traffic matrices. They extend their work with a decentralized algorithm that switches links OFF in an IP-based network to reduce the system energy consumption ([28]).

All these works identify the potential savings achieved by turning nodes OFF but ignore implementation obstacles such as the need for a mechanism that manages the network to limit packet losses and delays. In fact, it can be argued that any penalty in terms of QoS provided to users may have an indirect adverse affect on energy too. A user that has not been provided with sufficient QoS may need to repeat its use of the network and in doing so may incur additional energy costs. Thus in this paper, we examine the tradeoff between power and QoS by conducting experiments on a real, PC-based network topology and explore the challenges and issues that arise.

3. Trading Power Consumption against QoS

To investigate the effect of sleep modes on the nodes of a network and their consequence on QoS, we use a real testbed in which nodes have the sleep mode capability and more specifically the ACPI S3 suspend-to-RAM state [29], which is a low wake-latency sleep state, where only the RAM remains powered. We measure the power consumption of the network using the Watts up?.Net power meter [30]. The work we present aims to:

- Understand the drawbacks of introducing sleep modes in the network;

- Identify the required operating state of the network based on power consumption savings requirements and QoS constraints;
- Provide the basis for future mathematical models of the behaviour of such a system;
- Facilitate pricing techniques depending on QoS demands and the cost of power needed for these constraints to be guaranteed.

3.1. Proposed Mechanism

In order to achieve the best possible performance in a network where sleep modes are available, we built a mechanism consisting of the following modules:

- An enhanced neighbour-discovery mechanism for rapidly establishing whether a node's immediate neighbours are ON or OFF;
- A queue for storing packets at the previous hop if a node is suspended.

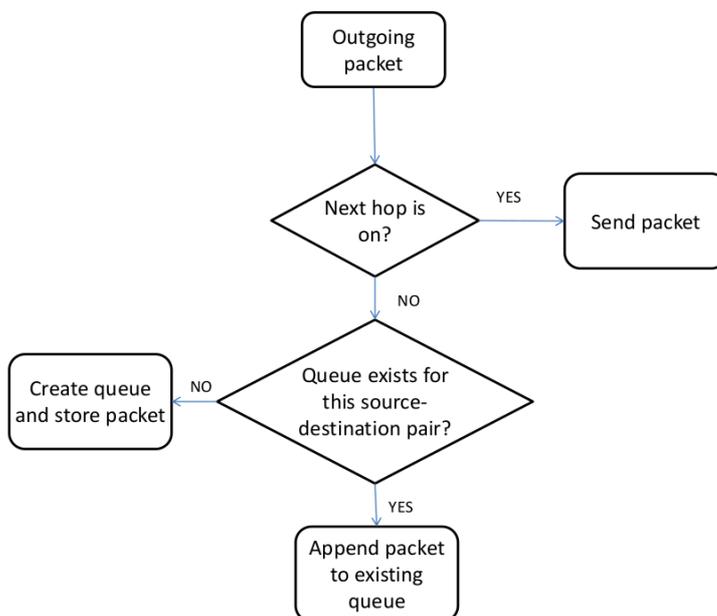
As soon as a given node A identifies that a set S of its neighbours are unavailable, it creates a queue for each unavailable neighbour. Packets of the same next-hop are queued until this neighbour becomes available again. As soon as the next hop wakes up, A receives hello packets from this node and after re-establishing the connection the packets in the corresponding queues are released towards their next hop. Note that a node that has packets in such queues may enter sleep mode. We address this by using the "suspend" capability, with which memory is the only remaining function that consumes energy, so as to avoid losing the packets in the queues. At each node, the queuing mechanism includes the following steps (see Figure1):

- When a packet is ready to be sent from a node, the state of the next hop is checked;
- If the next hop is ON then the packet is processed and sent out immediately;
- If the next hop is OFF the mechanism searches whether a storage queue exists for this hop.
 - if there is no queue, a new queue is created for this missing hop and the packet is stored;
 - if there is already a queue for this missing hop, the packet is simply placed at its end.

An enhanced neighbour discovery mechanism runs concurrently in each node. It is implemented by exchanging frequent "hello" packets between neighbours, to enable fast identification when the neighbour becomes unavailable by entering the energy-saving sleep mode. This mechanism therefore is responsible for:

- Sending frequent hello messages to inform the node's neighbours that it is ON;
- Keeping a state table of all its neighbours based on the hello messages it receives;
- Keeping the timestamp of the last hello message received from each neighbour;
- Removing a neighbour from the list if no hello messages have been received within a time-out interval;
- Re-adding a neighbour to the list when it wakes up from a sleep state;
- Processing the queue and sending out all the corresponding packets for a recently added neighbour;
- Informing the queuing mechanism which neighbours are ON (or not) based on the state tables of the neighbours.

Figure 1. Description of the queuing mechanism.



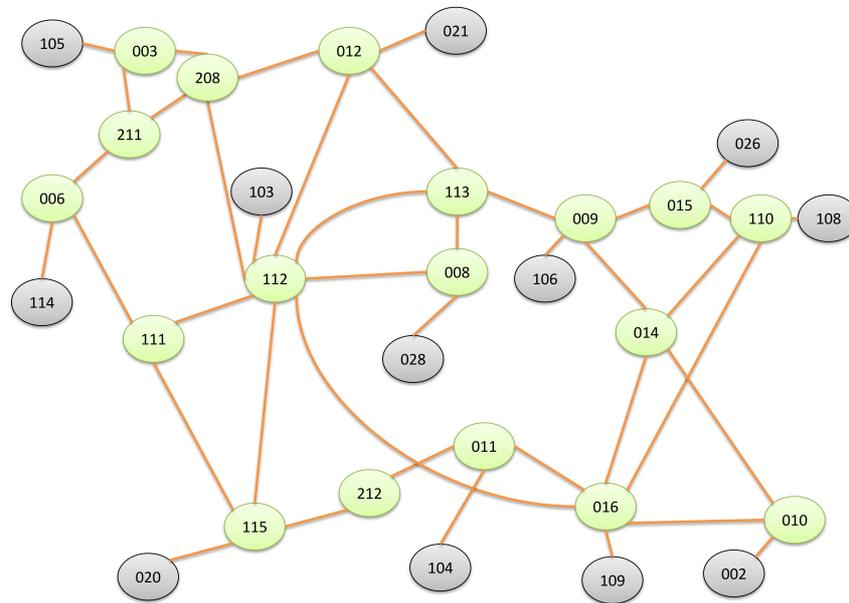
Intermediate nodes are controlled by an automated mechanism that decides when to suspend a node and when to turn it back ON again. Each node is controlled independently and SSH commands (pm-suspend) are sent to suspend the node, while wake-on-LAN messages are sent to wake the node from the suspended state [31].

4. Experiments

We evaluated the proposed mechanism with experiments conducted on a real 29-node testbed set up in a topology that mimics the CESNET2 network [32] as shown in Figure 2. In the experiments, as our focus is on the supporting wired network infrastructure, we do not differentiate based on how the users connect to it. For example, it could be a mobile user connecting through a base station to the network or connecting directly. Our focus is only on the duration that each user (mobile or not) is connected to the backbone network and the impact that an energy saving strategy of turning equipment OFF could have on the user’s QoS.

Our testbed consists of 29 PC-based physical routers running Ubuntu Linux 2.6.32-25. Their BIOS was configured to accept wake-on-LAN packets, and have the S3 suspend power state enabled so that they may be remotely put to sleep and waken up. The capacity of the links is 10 Mbps. Each user is represented by a source/destination pair generating continuous Poisson UDP traffic of 50 Kbps, to avoid congestion situations. The sources and destinations are always kept ON, while the intermediate nodes are put to sleep intermittently. The delay and packet loss of the traffic were measured as end-to-end values and the power consumption of the intermediate nodes was measured by using the *Watts up?.Net* power meter [30]. Each experiment lasted 200 seconds and was conducted 5 times. The results presented here are the average values of those runs.

Figure 2. Experimental topology. Grey nodes are sources/destinations and are always ON. Intermediate nodes can be put to sleep and their power consumptions are measured.



The network uses shortest path routing implemented as a Linux kernel module, and the queuing mechanism we have described is incorporated in the same module. The architecture used is not one of an interactive network with adaptive routing, but more of a store and forward network. We have chosen to use a predefined shortest-path routing and to store the packets until a path is re-established and not attempt to change paths when a node is OFF. This is because our aim is to study only the effect that turning nodes OFF has on the power consumption of a network, regardless of the routing protocol used. This also avoids cases such as the overloading of alternative paths and additional packet losses. Likewise, we have chosen to send UDP instead of TCP traffic so that the results will not be affected by the retransmission of packets.

Of course mechanisms similar to AODV [33] could be proved useful for future research. Such an on-demand approach for finding paths could be appropriate in cases where some of the nodes on the shortest path are asleep and alternative paths exist. Flood packets could be used to explore possible routes so that the most up-to-date information regarding the state of the nodes would be used.

Experiments are run with the auto-hibernation mechanism following a random pattern of turning OFF and ON the nodes. The duration that any node stays ON T_{on} and the duration that it stays in sleep mode T_{off} are selected randomly between two values $T_{on}^{min}, T_{on}^{max}$ and $T_{off}^{min}, T_{off}^{max}$ respectively. The average cycle time $T_{on}^{avg} + T_{off}^{avg} = T_{cycle}$ is kept constant for comparison. We report the results of seven experiments with the parameters shown in Table 1. During each experiment we measured the power consumption of the intermediate nodes, the packet loss and the average end-to-end delay experienced by the packets.

Table 1. Parameters used in each experiment.

	T_{on}^{min} (s)	T_{on}^{max} (s)	T_{off}^{min} (s)	T_{off}^{max} (s)
experiment 1	the nodes are always OFF			
experiment 2	5	10	5	30
experiment 3	5	15	5	25
experiment 3	5	20	5	20
experiment 5	5	25	5	15
experiment 6	5	30	5	10
experiment 7	the nodes are always ON			

In the first experiment the nodes are kept always ON, while in the last experiment the nodes are always OFF and are turned ON at the 200-th second. The minimum time that a node remains OFF (T_{off}^{min}) was chosen to be 5 seconds because this was the time it takes for a node to switch from normal operation into the suspended state.

In Figure 3 we examine further the sixth experiment and plot the measured power consumption over time. At the beginning of this experiment and after the 200-th second all nodes are ON, while during the experiment nodes turn OFF and ON at random. The measured end-to-end delays for all source/destination pairs in this experiment are shown in Figure 4 in the form of a histogram. It can be seen that most packets face very small delays, while as expected, there are some packets that face large delays due to being queued up in the network nodes when nodes on their path are in sleep mode. Thus, the power savings observed in this case (Figure 3) come at the expense of increased delay for some packets, while the majority of packets do not face any degradation in service.

Figure 3. Power consumption measurements of experiment 6.

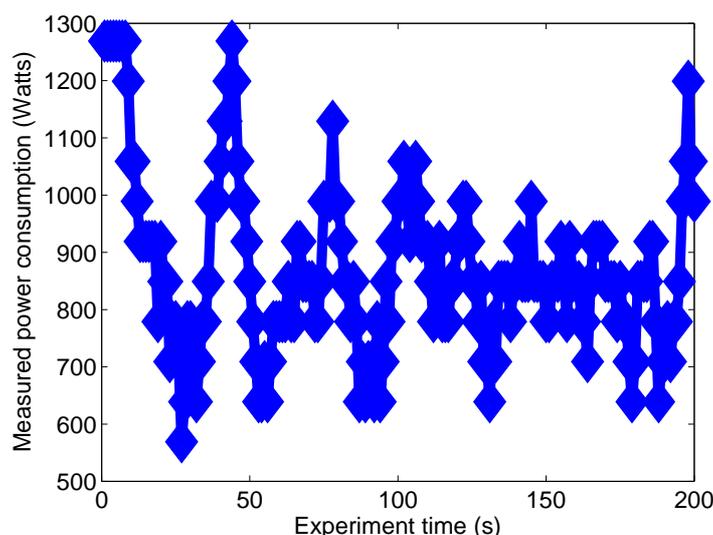
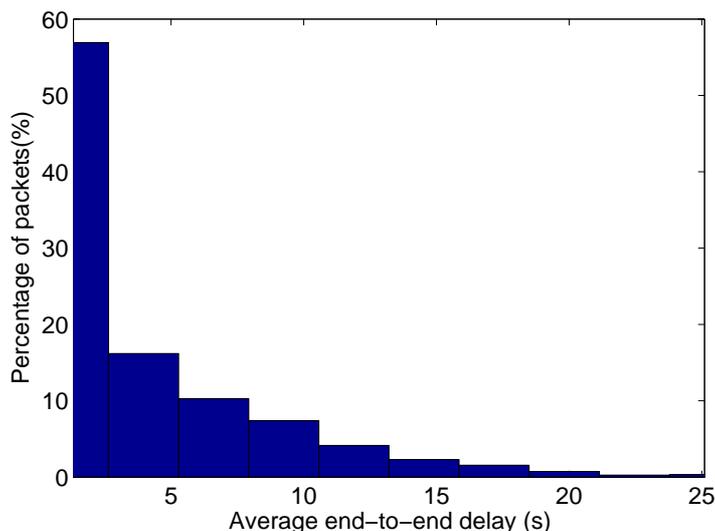


Figure 4. Histogram of delay measurements of experiment 6.



4.1. Effect of Sleep Modes on Packet Loss

We first examine the impact of putting nodes to sleep mode on packet loss, which is mainly due to the packets lost during the intervals between the instant that a neighbour goes to sleep and the time that it discovers that the neighbour is sleeping (no hello messages have been received during the time-out interval), after which it will start queuing the packets; packets may also be lost while the neighbour is entering the suspended state and is still replying to the hello messages but does not process the incoming traffic.

In Figure 5 the packet loss is plotted *versus* the proportion of time that nodes are ON, which is

$$R_{on} = \frac{T_{on}^{avg}}{T_{cycle}} \tag{1}$$

where $T_{on}^{avg} = (T_{on}^{max} + T_{on}^{min})/2$ is the average time the nodes are ON.

For the purposes of analysis, we take into consideration only experiments 2–6 and exclude the two extremes where the nodes are always OFF (experiment 1) or always ON (experiment 7). We observe that for the experiments 2–6, the packet loss is relatively stable, approximately 12%. This is expected, since we send traffic at a low rate to avoid network congestion and therefore packet loss is mainly due to the time it takes for a node to discover that a neighbour is asleep and start queuing the outgoing packets. Thus, packet loss seems to depend on the frequency of the hello messages and the timeout interval set. For comparison reasons, we repeated the experiments with $T_{cycle} = 50s$ and the same values of R_{on} (details described in Section 4.3) and as we can see in Figure 5 (right side) the packet loss is again relatively stable, at approximately 6%. This suggests that indeed the packet loss depends on the time it takes for a node to realise that a neighbour is asleep and is stable for the same T_{cycle} , while for larger T_{cycle} the packet loss is smaller as the total number of times the nodes are turned OFF are fewer. For our experiments, the hello messages between neighbours were sent every 1 s and the timeout was double that value, so as to avoid falsely identifying a neighbour as asleep and creating unnecessary queues. The packet losses could potentially be further reduced if the nodes signalled their neighbours just before they entered a sleep state.

Figure 5. Network average packet loss versus average ON rate R_{on} for $T_{cycle} = 25s$ and $T_{cycle} = 50s$.

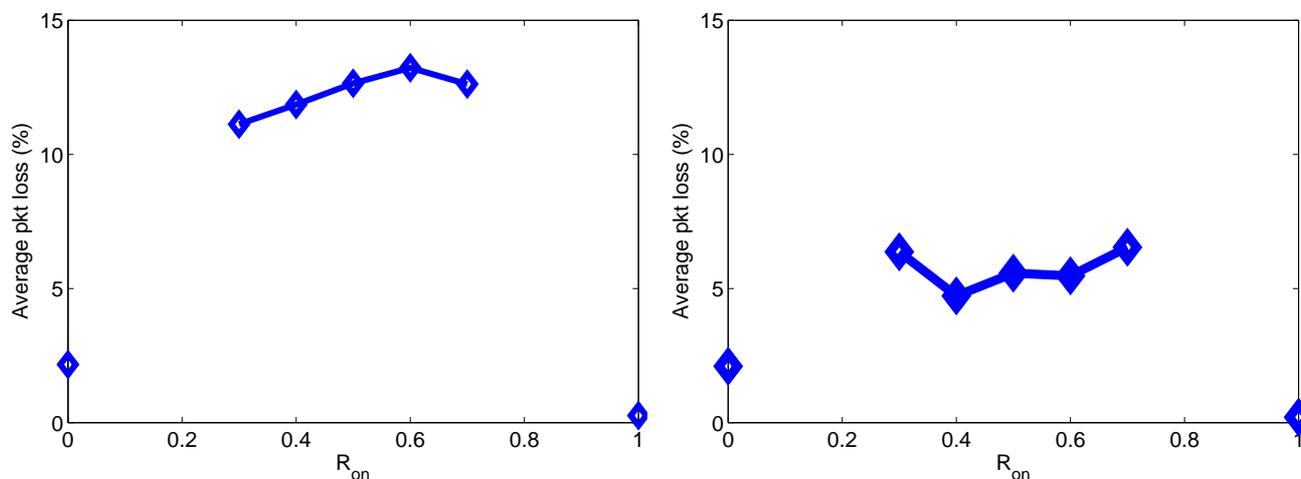


Figure 6 shows the average number of times that the nodes are turned ON/OFF in all experiments. The intermediate nodes turn ON/OFF approximately 125 times during each experiment, with the exception of experiments 1 and 7 where either all nodes are constantly ON or nodes are kept OFF until the end of the experiment. This was expected since we have kept the T_{cycle} constant. This allows consistency and enables us to compare the QoS values of the difference experimental cases and also justifies the relatively stable values of the packet losses mentioned before. Also, the measured fraction of packets that find an open path to the destination without queuing in any of the intermediate nodes are shown in Figure 7.

Figure 6. Total number of times that nodes were turned OFF/ON during each experiment.

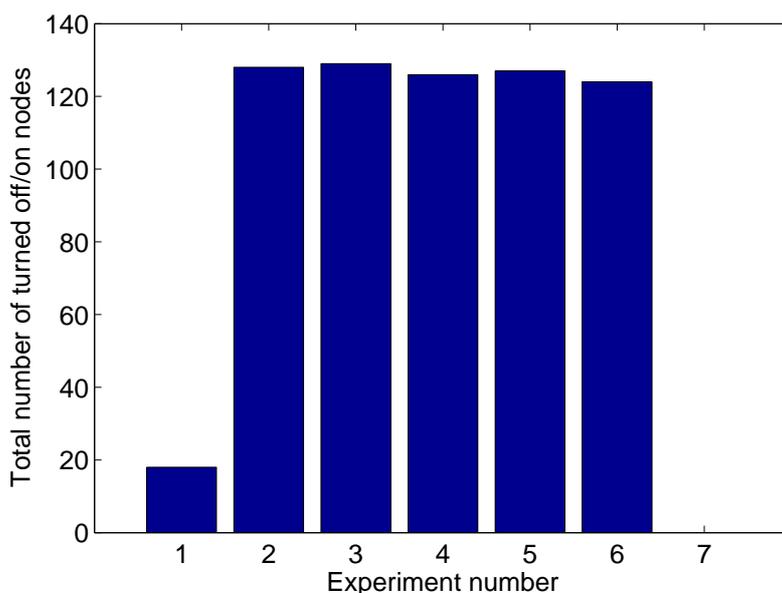
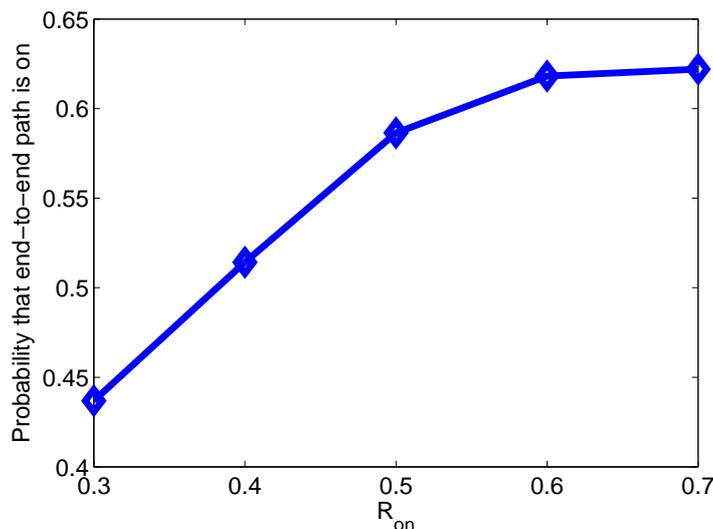


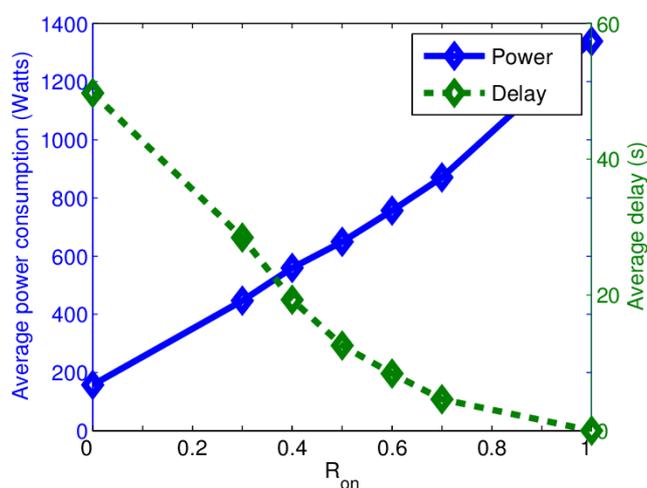
Figure 7. Probability that all nodes of a packet’s path are ON.



4.2. Power Consumption against Delay

The average value of the measured power against the R_{on} for all experiments is shown in Figure 8 (blue solid line). We observe that power consumption increases almost proportionally to R_{on} . Since R_{on} is the proportion of time that nodes stay ON, in order to save energy (smaller power consumption) R_{on} needs to be as small as possible. Obviously this comes at the price of a performance degradation since if nodes stay ON for less time (smaller R_{on}) the packet delay increases. This can be observed from the average delay curve in the same figure (green dotted line).

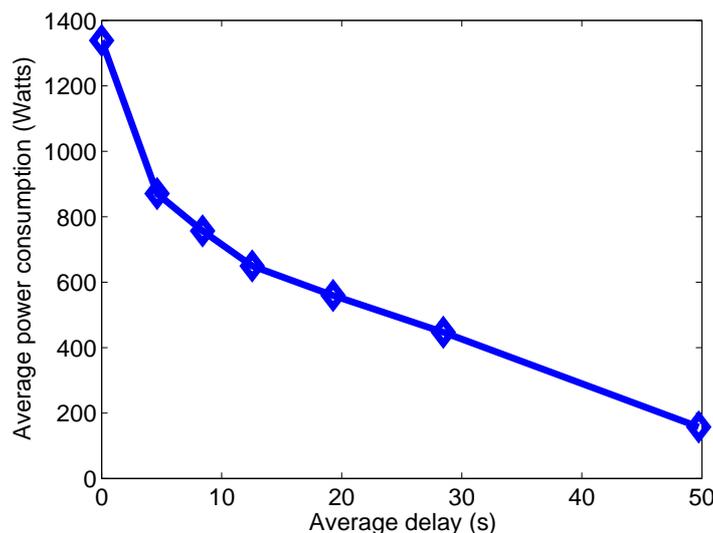
Figure 8. Average power consumption and average delay versus R_{on} .



Another way of presenting this relationship is by plotting the average measured value of power against the observed average packet delay for each experiment, shown in Figure 9. It is interesting to observe that the relationship between the average measured value of power consumption against the observed packet delay is not linear and the graph shows a knee. This is mainly due to the special case where all

nodes are ON ($R_{on} = 1$) and the delay is very small since there is no congestion in the network. Also, as the nodes remain turned OFF for longer periods (smaller R_{on} values) there are more packets being queued up in the nodes. Thus, there is an additional delay when the packets are waiting to be processed after the next hop is turned back ON.

Figure 9. Observed relationship between average power consumption and average delay for different values of R_{on} .



We do observe though a clear tradeoff between the power savings that can be achieved and the degradation in QoS that packets may experience presented in terms of delay. Power consumption is reduced by up to 88%, at the expense of increased delay that can be unacceptable for some of the traffic in the network. Through this relationship, one can choose the appropriate operating state of the network that offers the desired power savings and QoS constraints. For instance, in Experiment 6 ($R_{on} = 0.7$) we observe a 35% savings in network power consumption, while the end-to-end packet delay shown in Figure 4 is small for the most of the traffic.

4.3. The Impact of the ON/OFF Cycle

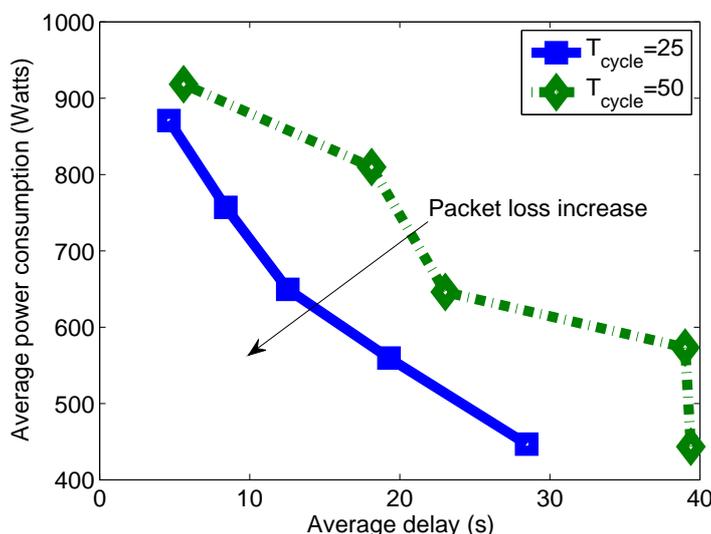
To investigate the effect that T_{cycle} has on the QoS of the packets, we repeat the experiments with $T_{cycle} = 50s$ and the same values of R_{on} . The parameters of this second set of seven experiments are shown in Table 2.

The results for packet loss are presented in Section 4.1 (Figure 5), where the packet loss is relatively stable at approximately 6%, smaller than when $T_{cycle} = 25s$. Figure 10 shows the measured power consumption *versus* the measured end-to-end delays for $T_{cycle} = 25s$ and $T_{cycle} = 50s$. In Figure 10 we can observe the system behaviour and the relative tradeoff between power consumption, delay and packet loss. Comparing between the two experiments, it can be seen that for larger T_{cycle} the delay increases. Thus, small T_{cycle} values result in smaller delays but larger packet losses.

Table 2. Parameters used in the set of experiment with $T_{cycle} = 50s$.

	T_{on}^{min} (s)	T_{on}^{max} (s)	T_{off}^{min} (s)	T_{off}^{max} (s)
experiment 1	the nodes are always OFF			
experiment 2	5	25	5	65
experiment 3	5	35	5	55
experiment 3	5	45	5	45
experiment 5	5	55	5	35
experiment 6	5	65	5	25
experiment 7	the nodes are always ON			

Figure 10. Average power consumption and average delay *versus* R_{on} for different T_{cycles} .



5. Conclusions

We have investigated the effect that the introduction of sleep modes on the energy consumption and QoS of a backbone network. To improve performance we have introduced the enhanced neighbour discovery in which queues at each node store outgoing packets whenever the next hop node is in sleep mode. The intermediate nodes are put to deep sleep or are waken independently of their traffic using a separate control mechanism to suspend or wake up a node. Experiments are reported for a real laboratory testbed using a shortest path routing protocol. The results can provide the basis for future modelling studies of such systems to identify optimal operating points for energy efficiency and QoS, and also to study techniques that can offer multiple-class differentiated energy and QoS characteristics.

References

- Gartner, I. Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO₂ Emissions, 2007. Available online: <http://www.gartner.com/newsroom/id/503867> (accessed on 14 May 2013).

2. Vereecken, W.; van Heddeghem, W.; Deruyck, M.; Puype, B.; Lannoo, B.; Joseph, W.; Colle, D.; Martens, L.; Demeester, P. Power consumption in telecommunication networks: Overview and reduction strategies. *Commun. Mag. IEEE* **2011**, *49*, 62–69.
3. Gelenbe, E.; Lent, R. Power-aware ad hoc cognitive packet networks. *Ad Hoc Netw.* **2004**, *2*, 205–216.
4. Sakellari, G. The cognitive packet network: A survey. *Comput. J.* **2009**, *53*, 268–279.
5. Berl, A.; Gelenbe, E.; Girolamo, M.; Giuliani, G.; de Meer, H.; Dang, M.; Pentikousis, K. Energy-efficient cloud computing. *Comput. J.* **2010**, *53*, 1045–1051.
6. Sakellari, G.; Loukas, G. A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing. *Simul. Modell. Pract. Theory* **2013**, in press.
7. Zhang, Y.; Chowdhury, P.; Tornatore, M.; Mukherjee, B. Energy efficiency in telecom optical networks. *IEEE Commun. Surveys Tutor.* **2010**, *12*, 441–458.
8. Bianzino, A.; Chaudet, C.; Rossi, D.; Rougier, J.L. A survey of green networking research. *Commun. Surv. Tutor. IEEE* **2012**, *14*, 3–20.
9. Gelenbe, E.; Silvestri, S. Reducing Power Consumption in Wired Networks. In Proceedings of the 24th IEEE International Symposium on Computer and Information Sciences (ISCIS '09), Guzelyurt, Northern Cyprus, 14–16 September 2009; pp. 292–297.
10. Gelenbe, E.; Silvestri, S. Optimisation of Power Consumption in Wired Packet Networks. In Proceedings of the 6th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE '09), Las Palmas, Gran Canaria, 23–25 November 2009; Springer: Berlin, Germany, 2009; Volume 22, pp. 717–729.
11. Adelin, A.; Owezarski, P.; Gayraud, T. On the Impact of Monitoring Router Energy Consumption for Greening the Internet. In Proceedings 11th IEEE/ACM International Conference on the Grid Computing (GRID), Brussels, Belgium, 24–29, October 2010; pp. 298–304.
12. Cianfrani, A.; Eramo, V.; Listanti, M.; Marazza, M.; Vittorini, E. An Energy Saving Routing Algorithm for a Green OSPF Protocol. In Proceedings of the INFOCOM IEEE Conference on Computer Communications Workshops, San Diego, CA, USA, 15–19 March 2010; pp. 1–5.
13. Panarello, C.; Lombardo, A.; Schembra, G.; Chiaraviglio, L.; Mellia, M. Energy Saving and Network Performance: A Trade-off Approach. In Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, Passau, Germany, 13–15 April 2010; pp. 41–50.
14. Gupta, M.; Singh, S. Greening of the Internet. *Comput. Commun. Rev.* **2003**, *33*, 19–26.
15. Sakellari, G.; Morfopoulou, C.; Mahmoodi, T.; Gelenbe, E. Using Energy Criteria to Admit Flows in a Wired Network. In Proceedings of the 27th International Symposium on Computer and Information Sciences (ISCIS 2012), Paris, France, 3–5 October 2012; pp. 63–72.
16. Gelenbe, E.; Morfopoulou, C. A framework for energy aware routing in packet networks. *Comput. J.* **2011**, *54*, 850–859.
17. Gelenbe, E.; Mahmoodi, T. Energy-Aware Routing in the Cognitive Packet Network. In Proceedings of the International Conference on Smart Grids, Green Communications, and IT Energy-aware Technologies (Energy 2011), Venice, Italy, 22–27 May 2011.
18. Gelenbe, E. Steps towards self-aware networks. *Commun. ACM* **2009**, *52*, 66–75.

19. Chabarek, J.; Sommers, J.; Barford, P.; Estan, C.; Tsiang, D.; Wright, S. Power Awareness in Network Design and Routing. In Proceedings of the IEEE INFOCOM 2008, Phoenix, AZ, USA, 13–18 April 2008; pp. 457–465.
20. Mahadevan, P.; Sharma, P.; Banerjee, S. A Power Benchmarking Framework for Network Devices. In Proceedings of IFIP Networking, Aachen, Germany, 11–15 May 2009.
21. Vasic, N.; Kostic, D. *Energy-Aware Traffic Engineering*; Technical Report for École polytechnique fédérale de Lausanne (EPFL): Lausanne, Switzerland, 2008.
22. Nedeveschi, S.; Popa, L.; Iannaccone, G.; Ratnasamy, S.; Wetherall, D. Reducing Network Energy Consumption via Sleeping and Rate-Adaptation. In Proceedings of the 5th Symposium on Networked Systems Design and Implementation, San Francisco, CA, USA, 16–18 April 2008; pp. 323–336.
23. Chiaraviglio, L.; Mellia, M.; Neri, F. Reducing Power Consumption in Backbone Networks. In Proceedings of IEEE International Conference on the Communications, ICC'09, Dresden, Germany, 14–18 June 2009; pp. 1–6.
24. Chiaraviglio, L.; Mellia, M.; Neri, F. Energy-Aware Backbone Networks: A Case Study. In Proceedings of IEEE International Conference on the Communications Workshops (ICC 2009), Dresden, Germany, 14–18 June 2009; pp. 1–5.
25. Chiaraviglio, L.; Ciullo, D.; Leonardi, E.; Mellia, M. How Much Can The Internet be Greened? In Proceedings of 2009 IEEE GLOBECOM Workshops, Honolulu, HI, USA, 30 November–4 December 2009; pp. 1–6.
26. Sanso, B.; Mellah, H. On Reliability, Performance and Internet Power Consumption. In Proceedings of the 7th International Workshop on Design of Reliable Communication Networks, Washington, DC, USA, 25–28 October 2009; pp. 259–264.
27. Bianzino, A.; Chaudet, C.; Larroca, F.; Rossi, D.; Rougier, J. Energy-Aware Routing: A Reality Check. In Proceedings of the IEEE GLOBECOM Workshops (GC Wkshps), Miami, FL, USA, 6–10 December 2010; pp. 1422–1427.
28. Bianzino, A.P.; Chiaraviglio, L.; Mellia, M.; Rougier, J.L. GRiDA: GReen Distributed Algorithm for energy-efficient IP backbone networks. *Int. J. Comput. Telecommun. Netw.* **2012**, *56*, 3219–3232.
29. Advanced Configuration and Power Interface Specification, Revision 5.0. Available online: <http://www.acpi.info/spec.htm> (accessed on 14 May 2013).
30. Wattsup. Available online: <https://www.wattsupmeters.com> (accessed on 14 May 2013).
31. Lieberman, P. *Wake-on-LAN technology*; Lieberman Software Corporation: Los Angeles, CA, USA, 2010.
32. CESNET2 Network. Available online: <http://www.ces.net/network/> (accessed on 14 May 2013).
33. Perkins, C.; Royer, E. Ad-hoc On-demand Distance Vector Routing. In Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99), New Orleans, LA, USA, 25–26 February 1999; pp. 90–100.