Routing and G-Networks to Optimise Energy and Quality of Service in Packet Networks

Erol Gelenbe and Christina Morfopoulou

Intelligent Systems and Networks Group
Dep. Electrical and Electronic Engineering
Imperial College, London SW7 2BT, UK
{e.gelenbe,c.morfopoulou}@imperial.ac.uk

Abstract. We formulate the problem of joint energy and quality of service (QoS) optimisation in packet networks and show how it can be formalised using the control capabilities inherent in G-network theory, which also includes the overhead due to control traffic. Using known energy consumption characteristics of network routers and of network link driver equipment, this approach leads to a polynomial time complexity gradient optimisation algorithm which seeks a judicious balance between the QoS experienced by the users and the overall energy consumption of the network.

Key words: Energy Optimisation, Wired Networks, Packet Routing, User QoS, Polynomial Time Complexity Optimisation Algorithm

1 Introduction

The ever increasing importance of wired packet networks as the transport substrate of most communication technologies creates the need for their greater energy efficiency. How to manage a wired packet network so as to enhance energy efficiency while respecting the users' QoS needs was studied in [1] and also for Cloud Computing in [2]. In this paper we examine the problem via an analytical approach. We first briefly review the literature on power management for wired networks and then we discuss relevant power consumption models. Our main result is a gradientr descent $O(N^3)$ time complexity algorithm that optimises power and QoS for a N-node network.

Energy savings in the Internet [3] can be achieved by routing modifications that aggregate traffic along a few routes and putting some nodes and devices to sleep.A network design problem where components can be powered on/off in combination with the solution of traffic assignment with flow-balance constraints is presented in [4]. An online technique to spread the load through multiple paths, based on the ability of nodes to adjust their operating rate is proposed in [5], and rate-adaptation for individual links is also examined in [6]. In [7] active links and routers are selected to minimize the power consumption via simple heuristics. In [8] a case study based on specific backbone networks is discussed,
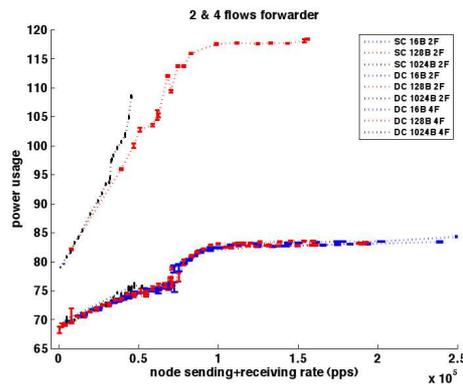
Fig. 1. Router power consumption as a function of packet rate[12]

and the potential overall energy savings in the Internet is estimated in [9]. The Cognitive Packet Network (CPN) [10, 11] for QoS is also considered in [1].

Another important set of problems that must be addressed when one attempts to design power management schemes for networks concern the proper choice of power consumption models for network components and sub-systems. In [4] a generic model for router power consumption is presented. Two widely used routers are measured in terms of system power demand with different configurations of line cards.Other work [6] focuses on the impact of the hardware processing rate and traffic on the power consumption. The power consumption model in this paper is based on measurements similar to those in Figure 1 [12].

2 Routing for Energy Management

In a network with N queues $\mathbf{N} = \{1, \; ... \; , N\}$,carrying a set of *user traffic classes.* $N+1$ denotes the "inexistent node" so that a packet that reaches node $N+1$ has simply exited the network or it has been lost. The term node is not restricted to the store and forward nodes of a network: a subset $\mathbf{R}$ is the usual store and forward nodes or routers, while the set $\mathbf{L}$ represents links which connect store and forward nodes, $\mathbf{N} = \mathbf{R} \cup \mathbf{L}$. Thus to travel from some store and forward router node to another such node, a packet will transit through the "other type of node" i.e. a link. This separation of the N nodes into the set of routers and the set of links has two advantages:

– It allows us to model separately the impact of routers and links (e.g. delay and loss) on QoS and on the *energy consumption*, and
– Secondly it allows us to explicitly represent packet re-routing as the modification of a path that goes (say) from router r via output link l, to a new path from router r via some other output to l'. Thus re-routing control can then viewed as taking place by an action on a packet in r that changes the next link that the packet must enter.

Links have a single predecessor node (which is a router) and a single successor node (also a router) and routers typically have multiple predecessor and successor nodes that are links. The two directions of a physical link are viewed as two distinct links, but they may be coupled because they may share the same power supply and the same hardware. Since router hardware will typically use "multi-cores", connections that are being concurrently processed by a router will often be handled by separate threads of execution, and packets within the same connection may be handled in first-come-first served order, while those of distinct connections may be processed in parallel.

Each user traffic class is denoted $k = U(s, d, \sigma)$ where (s, d) is a given source and destination router pair and σ is a QoS level that this traffic class expects to receive from the network. For the time being we will not specify the form that σ takes. We denote by $\lambda(i, k)$ the *external* arrival rate of packets of user class k to router i so that obviously $\lambda(i, k) = 0$ if $k = U(s, d, q)$ and $i \neq s$, while $\Lambda(i, k)$ is the total arrival rate of user traffic of class k at node i. Each class of users will, at any given instant of time, have a single path to its destination so that the probability that a packet of user class k travels from some node i to some node j is denoted by $P(i, k, j)$; if all packets of a given user class only travel over a single path at a time, then this quantity will be either equal to 1 or 0. Note also that if i is a router, the j is a link, while if i is a link, then j must be a router. Since user traffic is assumed to travel on a single path we obviously have that $\Lambda(s, k) = \lambda(s, k)$ when s is source router, and $\Lambda(i, k) = \Lambda(s, k)$ when i is not the source of the connection or user class k but i is a router or link that lies on the path of this traffic class.

We also have *control traffic classes* where each of these classes is in charge of selecting the next hop that a packet of a given user traffic class will take at some router. Since packets are re-routed by changing the selection of the outgoing link in a router, this selection will be signified by control packets. Thus control traffic classes are denoted (i, k) where i is a router and k is a *user* traffic class since a control class acts on a specific user class at some given router. Control traffic can travel over multiple hops just like any traffic class until it reaches the node where it is supposed to take action. The number of control traffic classes may be small if at a given instant a class of user traffic can only transit through a small number of network nodes.

A control packet of class (i, k) may move from router r to link l with probability $p((i, k), r, l)$ provided $i \neq r$ so that this particular control packet does not act at the router r to redirect traffic. This probability would be just 0 or 1 whenever there are no losses, simply because the output of a link is only connected to a single router. Similarly, the control packet moves from some link l to some router r, with probability $p((i, k), l, r)$.

The *control function* is represented by $Q(r, k, j)$, the probability that a user packet of class k at router r is *directed* by the control packet of type (i, k) to link j. $Q(r, k, j)$ is only defined at a router r for control class (r, k): in other words we need not specify how the control policy acts at a node where this particular control class is not empowered to act. Also, the control packets can only act at routers, so that $Q(i, k, j) = 0$ if $i \in \mathbf{L}$. We also assume that once a control packet

has acted at some router, then it is destroyed; in other words, each control packet can only act *once* on a single user packet at some specific router. If a control packet of class (r, k) arrives at router r when that router does not contain *any* user packets of class k, then the control packet is destroyed. For $r \in \mathbf{R}$:

$$\sum_{l \in \mathbf{L}} P(r, k, l) + P(r, k, N+1) = 1, \sum_{l \in \mathbf{L}} Q(r, k, l) = 1 \tag{1}$$

$$\sum_{l \in \mathbf{L}} p((j, k), r, l) + p((j, k), r, N+1) = 1, r \neq j \in \mathbf{R} \tag{2}$$

while if $l \in \mathbf{L}$:

$$\sum_{r \in \mathbf{R}} P(l, k, r) + P(l, k, N+1) = 1, \tag{3}$$

$$\sum_{r \in \mathbf{R}} p((i, k), l, r) + p((i, k), l, N+1) = 1, \ i \in \mathbf{R} \tag{4}$$

Note that there are no control packets classes of the form (l, k) where $l \in \mathbf{L}$. The control traffic classes may be seen in two ways:

– As flows of signaling information sent out from decision nodes, to routers where it may be necessary to re-route traffic, or
– As a mathematical representation of re-routing decisions. The arrival rate of control packets at some router may be used to represent the rate at which control decisions are made.

We assume that all user or control packets that travel *through* a node i have the same service rate μ_i at that node, but control packets act instantaneously when they act as control packets, rather than when they are simply transiting through a node and experience the usual queuing phenomenon. All packets are processed in first-come-first served mode in the nodes and links, so that there is no priority difference between user and control packets. On the other hand, when a control packet arrives at a node where it is supposed to act, it does this instantaneously on its "target" packet class, selecting the target packet of the appropriate class which is first in queue within its own class. If the router where this happens contains no packets of the target class, then the corresponding packet is destroyed. This model is a special case of G-Networks [13, 14, 15] with triggered customer movement, where control classes embody the triggers of the mathematical model [16], including multiple classes [17, 18]. Thus we are able to apply the corresponding theory, and the steady-state probability that the queue of node i contains at least one user packet of class k can be obtained as follows.

In many practical networks, each link only a single router at its input and output. However, in the notation below we continue to treat links as just another type of node, which is useful when links also represent data concentrators, dispatchers, buses or switching networks for distributed computer architectures and multiprocessors.

Let $\lambda^-(j, (i, k))$ be the *external* arrival rate of control traffic class $C(i, k)$ to router j, and such arrivals can only occur at routers. The *total arrival rate* of control traffic class (i, k) at router j, which will be computed below, is denoted

by $\Lambda^-(j,(i,k))$. This allows us to represent control traffic that may originate at different routers and act at the router where they originate, or they may act at other routers. The steady-state probability that a router or link contains at least one packet of user class k is given by:

$$q(r,k) = \frac{\Lambda_R(r,k)}{\mu_r + \Lambda^-(r,(r,k))}, \; if \; r \in \mathbf{R}, \tag{5}$$

We are assuming that each of the user classes are handled by separate queues in routers, e.g. when different user class queues within the same router are processed concurrently with a multicore architecture so that the queues are separate entities running in parallel. On the other hand, all packets within a link are handled in first-come-first-served order. The steady-state probability that the link l contains at least one packet of user class k is then

$$q(l,k) = \frac{\Lambda_L(l,k)}{\mu_l}, \; if \; l \in \mathbf{L}, \tag{6}$$

and the total arrival rates of user packets of class k to the routers and links are

$$\Lambda_R(r,k) = \lambda(r,k) + \sum_{l \in \mathbf{L}} q(l,k)P(l,k,r)\mu_l, \; if \; r \in \mathbf{R} \tag{7}$$

$$\Lambda_L(l,k) = \sum_{r \in \mathbf{R}} [P(r,k,l)q(r,k)\mu_r + \Lambda^-(r,(r,k))q(r,k)Q(r,k,l)], \; if \; l \in \mathbf{L} \tag{8}$$

while the arrival rate to router or link j of control traffic of class (i,k) is

$$\Lambda^-(j,(i,k)) = \lambda^-(j,(i,k)) + \sum_{l \in \mathbf{L}} p((i,k),l,j)c(l,(i,k))\mu_l, \; if \; i,j \in \mathbf{R} \tag{9}$$

$$= \sum_{r \in \mathbf{R}} p((i,k),r,j)K(r,(i,k))\mu_r, \; if \; i \in \mathbf{R}, j \in \mathbf{L}, i \neq r \tag{10}$$

where the steady-state probability that link l contains at least one control packet of class (i,k) is:

$$c(l,(i,k)) = \frac{\sum_{r \in \mathbf{R}} p((i,k),r,l)K(r,(i,k))\mu_r}{\mu_l}, l \in \mathbf{L}, i \in \mathbf{R} \tag{11}$$

and $K(r,(i,k))$ the steady-state probability that router r contains at least one control packet of class (i,k) for $r,i \in \mathbf{R}$ and $r \neq i$:

$$K(r,(i,k)) = \frac{\lambda^-(r,(i,k)) + \sum_{l \in \mathbf{L}} p((i,k),l,r)c(l,(i,k))\mu_l}{\mu_r} \tag{12}$$

Note that the steady-state probability that link l is busy is simply:

$$B(l) = \sum_{k \in \mathbf{U}} [q(l,k) + \sum_{i \in \mathbf{R}} c(l,(i,k))] \tag{13}$$

Assuming unbounded queue lengths, the average queue length at link l is

$$N(l) = \frac{B(l)}{1 - B(l)}, l \in \mathbf{L} \tag{14}$$

while consistent with the assumption that each category of packets, whether of user type or of control type, is handled in a separate queue at each router r, the average queue lengths at router r and $k \in \mathbf{U}$ are

$$N(r,k) = \frac{q(r,k)}{1 - q(r,k)} \tag{15}$$

$$N(r,(i,k)) = \frac{K(r,(i,k))}{1 - K(r,(i,k))}, i \neq r \tag{16}$$

We also define the probabilities that a user packet of class k, or a control packet of class (i,k), enters router r or link l

$$\pi(r,k) = \frac{\Lambda_R(r,k)}{\lambda^+(k)}, r \in \mathbf{R}, \pi(l,k) = \frac{\Lambda_L(l,k)}{\lambda^+(k)}, l \in \mathbf{L} \tag{17}$$

$$\pi(j,(i,k)) = \frac{\Lambda^-(j,(i,k))}{\lambda^-(i,k)}, j \in \mathbf{N}, i \in \mathbf{R}, i \neq j \tag{18}$$

where:

$$\lambda^+(k) = \sum_{r \in \mathbf{R}} \lambda(r,k) = \lambda(s,k) \tag{19}$$

is the total user traffic of class k in the network, s being its source router, and

$$\lambda^-(i,k) = \sum_{r \in \mathbf{R}} \lambda^-(r,(i,k)) \tag{20}$$

is the total control traffic of class (i,k). Using Little's formula [19, 20] the total average delay through the network for a user packet of class k is:

$$T(k) = \sum_{l \in \mathbf{L}} \pi(l,k) \frac{N(l)}{\Lambda_L(l,k)} + \sum_{r \in \mathbf{R}} \pi(r,k) \frac{N(r,k)}{\Lambda_R(r,k)} \tag{21}$$

while the total average delay experienced by a control packet of class (i,k) is:

$$T^-(i,k) = \sum_{l \in \mathbf{L}} \pi(l,(i,k)) \frac{N(l)}{\Lambda^-(l,(i,k))} + \sum_{r \in \mathbf{R}, r \neq i} \pi(r,(i,k)) \frac{N(r,(i,k))}{\Lambda^-(r,(i,k))} \tag{22}$$

The separation of nodes into routers $\mathbf{R}$ and links $\mathbf{L}$ allows us to model separately their power consumption. As indicated in [4], the power consumption of a router consists of a part that depends on the particular chassis type, and another determined by the line cards in the route; the power consumption includes the amount needed to keep it on, the operating for route changes, and that which is needed for processing individual packets. Let Λ_i^+ be the total traffic of user packets entering node i, while Λ_i^- is the total control transiting that node. They are given by

$$\Lambda_i^+ = \sum_{k \in \mathbf{U}} \Lambda_R(i,k) \; if \; i \in \mathbf{R} \tag{23}$$

$$= \sum_{k \in \mathbf{U}} \Lambda_L(i,k) \; if \; i \in \mathbf{L} \tag{24}$$

$$\Lambda_i^- = \sum_{j \neq i} \sum_{k \in \mathbf{U}} \Lambda^-(i, (j, k)) \qquad (25)$$

and the total traffic of packets Λ_i transiting through a node i will be

$$\Lambda_i = \Lambda_i^+ + \Lambda_i^- \qquad (26)$$

The measurements reported in Figure 1 [12] for two distinct machines used as routers and different fixed packet lengths, show that for older single core technology (curves above) the power consumption increases monotonically with the rate at which *packets* are processed in the router. The curves below show similar results for a more recent multicore technology with much lower power consumption, and have a distinct step upwards when an additional core kicks in as packet rate increases. In all cases *packet length* has little effect on power consumption. Thus, we use the following power consumption formula for a router

$$P_i = \alpha_i + g_R(\Lambda_i) + c_i \sum_{k \in \mathbf{U}} \Lambda^-(i, (i, k)), i \in \mathbf{R} \qquad (27)$$

where α_i corresponds to the router's static power consumption, $c_i > 0$ is a constant, $g_R(.)$ is an increasing function of the packet processing rate, as in Figure 1, while c_i is a proportionality constant related to the amount of processing being carried out in the router for re-routing control. The link power will depend on the traffic rate in bytes or bits per second (rather than packets per second), and includes the needs for operating the interface with the router, and for transmitting data on the line. Additionally, one could include the power consumed for propagating or "repeating" data on the line, but this may be negligible [21]. The link power model that we propose is then:

$$P_i = \beta_i + g_L(\Lambda_i), i \in \mathbf{L} \qquad (28)$$

where β_i corresponds to the static power consumption, and $g_L(.)$ is an increasing function. Thus (27) and (28) allow us to examine the impact of routing decisions via changes in traffic rates, and also to evaluate the effect of putting different routers or links "to sleep". Since link interfaces can consume up to 40% of the overall router power, and because they can be put to sleep or woken up much more rapidly than a whole router, one can benefit from just turning links off.

The users' QoS needs are typically expressed in terms of packet delay, probability of loss, jitter that depend on the congestion at routers and links, which depend on the probabilities that the nodes or links are busy. Thus the $q(i, k)$, $c(l, (i, k))$ and $K(r, (i, k))$ are the key quantities we will use to obtain the QoS metrics.

The average overall network packet delay is given by:

$$\overline{T_N} = \sum_k \frac{\lambda^+(k)}{\Lambda_T^+} T(k) \qquad (29)$$

where $\lambda^+(k)$ is given by equation 19, $T(k)$ is given by (21) and $\Lambda_T^+ = \sum_k \lambda^+(k)$.

2.1 Gradient Descent Optimisation

Routing optimization can be expressed as the minimization of a function f that includes both the Network Power Consumption and the Average Delay:

$$\text{Minimize } f = c \sum_i P_i + \overline{T_N} \text{ by selecting control parameters } Q(x, m, y) \qquad (30)$$

Since we are interested in gradual improvements in the presence of ongoing flows, we compute the partial derivative of f :

$$\frac{\partial f}{\partial Q(x, m, y)} = c \sum_{i \in \mathbf{N}} \frac{\partial P_i}{\partial Q(x, m, y)} + \frac{\partial \overline{T_N}}{\partial Q(x, m, y)} \qquad (31)$$

where we use (27), (28) and (29): Since $\frac{\partial P(i,k,j)}{\partial Q(x,m,y)} = \frac{\partial p((i,k),j,n)}{\partial Q(x,m,y)} = 0$ we calculate $\frac{\partial q(r,k)}{\partial Q(x,m,y)}$ and $\frac{\partial q(l,k)}{\partial Q(x,m,y)}$. Define $h(i,j) = 1$ if there is a physical connection from node i to j and $h(i,j) = 0$ otherwise. Note that for evaluating $\frac{\partial \mathbf{X}}{\partial Q(x,m,y)}$ we need only consider cases where $h(x,y) = 1$. Define the vector $\mathbf{q}_k = (q(1,k), q(2,k), .., q(N,k))$ and the $N \times N$ matrices:

$$\mathbf{A}_k = [A_k(l,r)], A_k(l,r) = \frac{P(l,k,r)}{\mu_r + \Lambda^-(r, (r,k))} \qquad (32)$$

$$\mathbf{D}_k = [D_k(r,l)], D_k(r,l) = [P(r,k,l)\mu_r + \Lambda^-(r, (r,k))Q(r,k,l)) \qquad (33)$$

$$\mathbf{B}_k = [B_k(l,r)], B_k(l,r) = \mu_l P(l,k,r) \qquad (34)$$

$$\mathbf{C}_k = [C_k(r,l)], C_k(r,l) = \frac{[P(r,k,l)\mu_r + \Lambda^-(r, (r,k))Q(r,k,l)]}{\mu_l[\mu_r + \Lambda^-(r, (r,k))]} \qquad (35)$$

and the $1 \times N$ row vectors :

$$\mathbf{N}(l) = 1/\mu_l \qquad (36)$$

$$\mathbf{H}_k^{xmy}(l) = \begin{cases} \Lambda^-(x, (x,k))q(x,k) & k = m, l = y \\ -h(x,l)\Lambda^-(x, (x,k))q(x,k) & k = m, l \neq y \\ 0 & \text{otherwise} \end{cases} \qquad (37)$$

Then after some calculations we get:

$$\frac{\partial \mathbf{q}_k(r)}{\partial Q(x, m, y)} = \frac{\partial \mathbf{q}_k(r)}{\partial Q(x, m, y)} \mathbf{D}_k \mathbf{A}_k + \mathbf{H}_k^{xmy} \mathbf{A}_k \qquad (38)$$

$$\frac{\partial \mathbf{q}_k(l)}{\partial Q(x, m, y)} = \frac{\partial \mathbf{q}_k(l)}{\partial Q(x, m, y)} \mathbf{B}_k \mathbf{C}_k + \mathbf{H}_k^{xmy} \mathbf{N} \qquad (39)$$

where $r \in \mathbf{R}$ and $l \in \mathbf{L}$. Thus, equations (38) and (39) can be written as:

$$\frac{\partial \mathbf{q}_k}{\partial Q(x, y, m)} = \frac{\partial \mathbf{q}_k}{\partial Q(x, y, m)} \mathbf{W}_k + \boldsymbol{\gamma}_k^{xmy} \qquad (40)$$

where the matrix $\mathbf{W}_k$ and the vector $\boldsymbol{\gamma}_k^{xmy}$ are given by

$$W_k(i,j) = \begin{cases} \sum_{l \in \mathbf{L}} D_k(i,l)A_k(l,j) & i,j \in \mathbf{R} \\ \sum_{r \in \mathbf{R}} B_k(i,r)C_k(r,j) & i,j \in \mathbf{L} \end{cases} \qquad (41)$$

$$\gamma_k^{xmy}(n) = \begin{cases} \sum_{l \in \mathbf{L}} H_k^{xmy}(l) A_k(l,n) & n \in \mathbf{R} \\ H_k^{xmy}(n) N(n) & n \in \mathbf{L} \end{cases} \tag{42}$$

So,

$$\frac{\partial \mathbf{q}_k}{\partial Q(x,y,m)} = \boldsymbol{\gamma}_k^{xmy}(\mathbf{I} - \mathbf{W}_k)^{-1} \tag{43}$$

where $\mathbf{I}$ is the $N \times N$ identity matrix. Using (43) we can calculate $\partial f / \partial Q(x,m,y)$ from equation (31). Note that the matrix inversion is of time complexity $O(N^3)$.

The gradient descent algorithm to obtain the parameters $Q(i,k,j)$ that reduce the cost function at a given operating point of the network $\underline{X} = [\underline{\lambda}, \underline{\lambda}^-, \underline{\mu}, \underline{P}^+, \underline{p}]$ is determined by its nth computational step:

$$Q_{n+1}(i,k,j) = Q_n(i,k,j) - \eta \frac{\partial f}{\partial Q(i,k,j)}|_{Q(i,k,j)=Q_n(i,k,j)} \tag{44}$$

where $\eta > 0$ is the "rate" of the gradient descent and the partial derivative is computed with the nth updated values of the weights. The steps of the learning algorithm are then:

1. Initialise all the values $Q(i,k,j)$ and set $\eta > 0$.
2. Solve the $U \times N$ nonlinear equations (5)-(12) to obtain the $q(i,k)$.
3. Solve the $U \times N$ linear equations (43) using the $q(i,k)$.
4. Using the steps 2 and 3 update $Q(i,k,j)$ (44)

3 Conclusion

Technological means to greater energy efficiency such as the Smart Grid, distance learning and E-work, depend increasingly on computer networks and information technology: while they reduce energy consumption in transportation (for instance), but increase the need for energy to run networks and ICT systems. Thus, we must develop seek energy efficient ways to run ICT systems. Our work seeks techniques that use routing control in a network as a means to reduce energy consumption while remaining aware of QoS considerations. The proposed approach is based on network analysis methods that dynamically distribute traffic so as to minimise a cost function that includes energy and QoS. G-Networks are mathematical models that incorporate both the user traffic and the traffic used to control packet routing. Based on G-networks, we have developed a model that incorporates routers and links as separate queuing servers, and the model is also able to incorporate the overhead of the control traffic as well as the ordinary payload traffic in the network. A gradient based algorithm for progressive traffic re-routing is developped to minimise a cost function which incorporates both the energy consumption and the QoS. The algorithm is shown to be of $O(N^3)$ time complexity in the size of the network. In future work we expect to present an experimental implementation of energy aware routing based on the modelling and analysis techniques developped in this paper.

4 Acknowledgment

References

1. E. Gelenbe and S. Silvestri, "Reducing power consumption in wired networks," in *24th International Symposium on Computer and Information Sciences (ISCIS'09)*, North Cyprus, 14-16 September 2009, (IEEE Digital Library).
2. 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*, accepted for publication 2009.
3. M. Gupta and S. Singh, "Greening of the internet," *Computer communication review*, vol. 33, no. 4, 19–26, 2003.
4. J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008, 457 –465.
5. N. Vasic and D. Kostic, "Energy-aware Traffic Engineering," EPFL, Tech. Rep., 2008.
6. S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation.* Berkeley, CA, USA: USENIX Association, 2008, 323–336.
7. L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *Communications, 2009. ICC '09. IEEE International Conference on*, June 2009, 1 –6.
8. ——, "Energy-aware backbone networks: A case study," in *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, June 2009, 1 –5.
9. L. Chiaraviglio, D. Ciullo, E. Leonardi, and M. Mellia, "How much can the internet be greened?" in *GLOBECOM Workshops, 2009 IEEE*, 30 2009-Dec. 4 2009, 1 –6.
10. E. Gelenbe, "Cognitive packet network," *U.S. Patent 6,804,201*, October 11, 2004.
11. ——, "Steps towards self-aware networks," *Communications of the ACM*, .52(7):66-75, July 2009.
12. R. Lent, "Power measurements of processors for routing," *to appear*, 2010.
13. E. Gelenbe, "G-networks: An unifying model for queuing networks and neural networks," *Annals of Operations Research*, vol. 48, no. 1-4, 433–461, 1994.
14. ——, "G-networks with signals and batch removal," *Probability in the Engineering and Informational Sciences*, vol. 7, 335–342, 1993.
15. J.-M. Fourneau and E. Gelenbe, "Flow equivalence and stochastic equivalence in g-networks," *Computational Management Science*, vol. 1, no. 2, 179–192, 2004.
16. E. Gelenbe, "G-networks with instantaneous customer movement," *Journal of Applied Probability*, 30 (3), 742–748, 1993.
17. J.-M. Fourneau, E. Gelenbe, and R. Suros, "G-networks with multiple classes of negative and positive customers," *Theoretical Computer Science*, 155, 141–156, 1996.
18. E. Gelenbe and A. Labed, "G-networks with multiple classes of signals and positive customers," *European J. of Operations Research*, 108 (2), 293–305, 1998.

19. E. Gelenbe and I. Mitrani, *Analysis and Synthesis of Computer Systems*. Imperial College Press, World Scientific, 2010.
20. E. Gelenbe and G. Pujolle, *Introduction to Queueing Networks*, 2nd ed. John Wiley & Sons Ltd, 1998.
21. J. Baliga, K. Hinton, and R. S. Tucker, "Energy consumption of the internet," in *Optical Internet, 2007 & 2nd Australian Conf. on Optical Fibre Tech. COIN-ACOFT 2007. Joint International Conf. on*, June 2007, 1 –3.