A Framework for Energy Aware Routing in Packet Networks

Erol Gelenbe and Christina Morfopoulou

Intelligent Systems and Networks Group, Department of Electrical and Electronic Engineering, Imperial College, London SW7 2BT, UK
Email: e.gelenbe,c.morfopoulou@imperial.ac.uk

We examine on-line routing algorithms for wired packet networks to improve power consumption and quality of service (QoS). For an N-node network we derive an on-line control scheme which operates in $O(N^3)$ time complexity. Then we obtain a very low-complexity approach using load-balancing that reduces the overall cost including power consumption and network QoS.

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

Received 00 January 2009; revised 00 Month 2009

1. INTRODUCTION AND RELATED WORK

The increasing importance of wired packet networks as the transport substrate on which many communication technologies are converging, also increases the importance of making these networks efficient in different aspects. A major need for greater efficiency is in the power consumption of such networks, which can be improved in several ways. The energy efficiency of all the equipment that is used in networks is of primary concern and equipment manufacturers as well as chip designers are increasingly aware of this problem. One could also examine the manner in which the network topology can be designed so as to optimise for energy efficiency within a desired framework of traffic load and quality of service (QoS) constraints.

Another approach which comes in addition to these considerations is the opportunity to manage a given store-and-forward wired packet network, which has certain installed routers and network interfaces as well as other equipment, so as to enhance its energy efficiency while respecting the users' QoS needs and the prior service level agreements (SLA). This question has been explored experimentally [1] and discussed in the framework of Cloud Computing [2]. In this paper we examine the problem via an analytical approach.

In the sequel, we first briefly review the literature related to power management in wired and wireless networks. We then discuss power consumption models for wired packet networks. This is followed by a formalisation of a theoretical framework to design algorithms that optimise power consumption and quality of service (QoS). For an N-node network we show that the optimisation of a composite power and QoS based cost function can run in $O(N^3)$ time complexity. But due to the complexity of this approach, we also show that load-balancing will reduce a similar cost function.

1.1. Related work

Early work suggesting the idea of energy savings in the Internet [3] 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. A network-wide approach (coordinated sleeping) as well as a link layer approach (uncoordinated sleeping) were discussed and the possible effects on routing protocols were examined. Later work [4] designed and evaluated an energy saving algorithm for Ethernet links using local data to make sleeping decisions. In [5], a network design problem where components can be powered on/off in combination with the solution of a multicommodity network-flow problem for traffic assignment with flow-balance constraints is solved using an offline optimization method that estimates the potential power savings. An energy-aware online technique is proposed in [6] 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 of the hardware and the ability of nodes to automatically adjust their operating rate to their utilization. Rate-adaptation for individual links is also examined in [7] 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 [8] the authors select the active links and routers to minimize the power consumption via simple heuristics that approximately solve a corresponding NP-hard problem. In [9] a case study based on specific backbone networks is discussed, and an estimate of the potential overall energy savings in the Internet is presented in [10]. In [1] the reduction of power consumption in wired networks is studied in the presence of users' QoS constraints and experiments with dynamic traffic management in conjunctions with the turning on/off of link drivers and/or routers are discussed. The use of the Cognitive Packet Network (CPN) [11, 12] routing protocol that would use energy awareness in conjunction to QoS is also considered. Energy efficiency is examined in the Cloud Computing context in [2] where an overview of existing energy saving techniques in computer hardware and network infrastructure is presented for such environments.

In contrast to wired networks where power management has received relatively less attention, much work has been presented for wireless networks. In particular, wireless sensor networks may be limited by battery power, so energy minimization is of crucial importance. One of the approaches proposed is Topology Control (TC) [13, 14] so as to dynamically adjust the radio transmission power and hence the range of each node while also preserving the connectivity of each potential source-destination pair, also controlling wireless interference and network capacity. In [15] TC with QoS constraints is examined. Other approaches consider that some nodes can be in a sleeping mode and the objective is to find a subnetwork of active nodes which respect the connectivity constraint at any time. In [16] it is indicated that the radio transceiver, which is the dominant energy consumer within a sensor, consumes almost the same amount of energy in transmit, receive and idle mode. Therefore switching off only the transceiver can result in meaningful energy savings. The possibility of turning nodes off is also addressed in [17]. The authors examine two algorithms for energy conservation which run on top of existing ad hoc routing protocols. It is noted that there are costs associated with turning off nodes including increased latency and packet loss, and therefore algorithms should be designed so as to reach a trade-off between energy consumption and QoS. In [18] an energy efficient routing algorithm for ad hoc networks is presented which extends CPN [11] which is a fast adaptive routing algorithm that uses smart packets for QoS based path finding. The resulting Ad-Hoc CPN (AHCPN) uses a metric called path availability representing a quantity that is proportional to estimated remaining lifetime of battery power at the nodes which compose a path. The combination of path availability and delay in a single routing criterion provides packets with higher QoS paths and also encourages them to use less power.

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 [5] 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. The measurements indicate that the base system is the largest power consumer, so it is best to minimize the number of chassis at a given PoP and maximize the number of cards per chassis. On the other hand, other work [7] 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 [19] for routers, and Figure 2 for links [6].

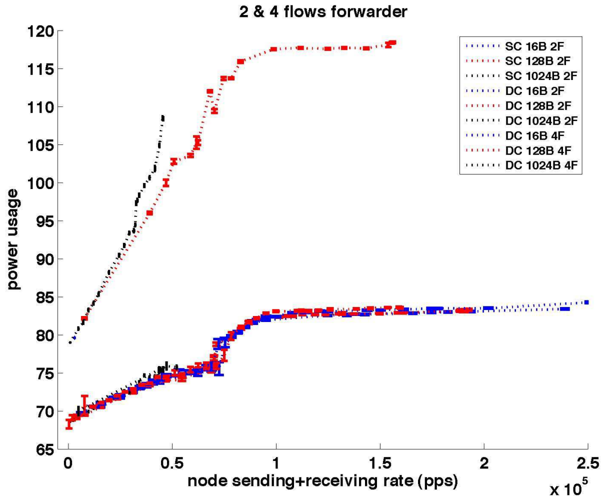

FIGURE 1. Router power consumption in Watts as a function of packet rate in packets per second [19]

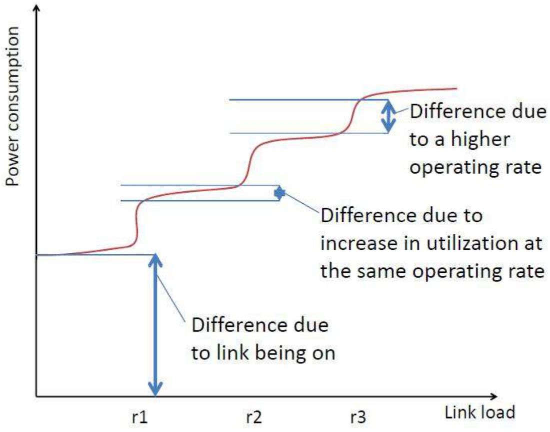

FIGURE 2. Proposed power consumption in Watts as a function of traffic rate in bytes per second [6]

2. ROUTING FOR ENERGY MANAGEMENT

Consider a network with N queues denoted $\mathbf{N} = \{1, \ldots, N\}$ which is carrying a set of *user traffic classes*.

Let $N+1$ be the "inexistent node" or the outside of the network. Thus a packet that reaches node $N+1$ has simply exited the network or it has been lost. Here the term node is not restricted to the store and forward nodes of a network: a subset $\mathbf{R}$ of these nodes will be the usual store and forward nodes or routers, while the remaining set of nodes $\mathbf{L}$ will represent links which connect store and forward nodes, i.e. $\mathbf{N} = \mathbf{R} \cup \mathbf{L}$. 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 be 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 (which is also a router), and routers typically have multiple predecessor and successor nodes that are links. In our notation 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 some of the same hardware. Since router hardware will typically use "multicores", 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.

In addition to user traffic, 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. In other words, packets that belong to a control class are in charge of telling a given router where the packets of a given user traffic class must go. Since packets are re-routed by changing the selection of the outgoing link in a router, this selection will be signified by these 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. However, 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 actual number of control traffic classes will be small because at a given instant of time a class of user traffic will only transit through a small number of network nodes.

A control packet of class (i, k) may move from some router r to some link l with probability $p((i, k), r, l)$ provided $i \neq r$ so that this particular control packet cannot act at the router r to redirect traffic. Similarly, the control packet moves from some link l to some router r, with probability $p((i, k), l, r)$; Note that 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.

The *control function* exercised by the control packet will be represented by $Q(r, k, l)$ which is the probability that a user packet of class k at router r is *directed* by the corresponding control packet of type (r, k) to the link l. Note that $Q(r, k, l)$ is only defined at a router r for the 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. Furthermore, 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 again destroyed. For $r \in \mathbf{R}$:

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

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

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

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

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

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

- As physical flows of signaling information in the form of specific packets, that are sent out from certain decision nodes, to routers where it may be necessary to re-route traffic, or
- As a virtual and mathematical representation of re-routing decisions. Thus the arrival rate of control packets at some router may be used to represent the rate at which control decisions are made at this router.

We also 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 when they experience the usual queuing phenomenon. Finally, all packets are processed in first-come-first-served mode both in the nodes and links, so that there is no priority difference for transiting purposes between the user and control packets. On the other hand, when a control packet arrives at a node where it is supposed to take the control action, it does this instantaneously on its "target" packet class, selecting the target packet (i.e. of the appropriate class) which is first in queue within its own class. If the router where this is supposed to happen contains no packets of the target class, then the corresponding packet is destroyed. The model that we use is a special case of G-Networks [20, 21, 22] with triggered customer movement, where the control classes embody the triggers of the mathematical model [23], including multiple classes [24, 25]. Thus we are able to apply the corresponding theory, and in particular we know that 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 has a single router at its input, and also each link has a single router at its output. However, in the notation below we continue to treat links as just another type of node, which can be of interest when links can also be used to represent data concentrators and dispatchers, buses or switching networks for distributed computer architectures and multiprocessors.

We denote by $\lambda^-(j,(i,k))$ the *external* arrival rate of control traffic class (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:

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

where we are assuming that each of the user classes are handled by separate queues in routers; recall that we assume that these different user class queues within the same router are processed concurrently with the help of a multicore architecture so that the queues can indeed be viewed as 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 given by:

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

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

$$\begin{aligned}
\Lambda_R(r,k) &= \lambda(r,k) + \sum_{l \in \mathbf{L}} q(l,k) P(l,k,r) \mu_l \\
&= \lambda(r,k) + \sum_{l \in \mathbf{L}} P(l,k,r) \Lambda_L(l,k),\ if\ r \in \mathbf{R} \qquad (7) \\
\Lambda_L(l,k) &= \sum_{r \in \mathbf{R}} [P(r,k,l) q(r,k) \mu_r \\
&+ \Lambda^-(r,(r,k)) q(r,k) Q(r,k,l)],\ if\ l \in \mathbf{L} \qquad (8)
\end{aligned}$$

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

$$\begin{aligned}
\Lambda^-(j,(i,k)) &= \lambda^-(j,(i,k)) + \sum_{l \in \mathbf{L}} p((i,k),l,j) c(l,(i,k)) \mu_l, \\
&\quad if\ i,j \in \mathbf{R} \qquad (9) \\
&= \sum_{r \in \mathbf{R}} p((i,k),r,j) K(r,(i,k)) \mu_r, \\
&\quad if\ i \in \mathbf{R}, j \in \mathbf{L}, i \neq r \qquad (10)
\end{aligned}$$

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

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

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 $r \neq i$:

$$\begin{aligned}
K(r,(i,k)) &= \frac{\lambda^-(r,(i,k)) + \sum_{l \in \mathbf{L}} p((i,k),l,r) c(l,(i,k)) \mu_l}{\mu_r} \\
&= \frac{\lambda^-(r,(i,k)) + \sum\limits_{l \in \mathbf{L}} \sum\limits_{j \in \mathbf{R}} p((i,k),l,r) p((i,k),j,l) K(j,(i,k)) \mu_j}{\mu_r} \qquad (12)
\end{aligned}$$

Finally, let us point out that the steady-state probability that link l is busy is simply:

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

We now derive the average queue lengths at routers and links. Assuming that queue lengths are unbounded, we note that the average queue length at link l is given by:

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

while the average queue lengths at the routers will be consistent with the preceding discussion and assume that each category of packets, whether of user type or of control type, will be handled in a separate queue at each router r, so that 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 entering the network, s being the source router of class k, 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 [26, 27] 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 [5], the power consumption of a router consists of a part that depends on the particular chassis type, and another part determined by the line cards in the router, and power consumption includes the amount needed to keep it on, the operating for route changes, and that which is needed for processing individual the packets. Let Λ_i^+ be the total traffic of user packets entering node i, while Λ_i^- is the total control transiting that node, and they are given by:

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

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

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

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

$$\Lambda_i = \Lambda_i^+ + \Lambda_i^- \tag{26}$$

The measurements reported in Figure 1 [19] 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 the power consumption, which is dominated by the number of packets processed per unit time. As a result, we will use the following power consumption formula for a router:

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

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 consumption shown in Figure 2, depends 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 [28]. The link power model that we propose is then:

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

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 may also be used 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 already benefit from just turning links off.

The users' QoS needs are typically expressed in terms of packet delay, probability of loss, jitter, and similar metrics 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 other QoS metrics. The average overall network packet delay is given by:

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

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

The routing optimization can now be expressed as the minimization of a function f that includes both the

Network Power Consumption and the Average Delay:

$$\text{Minimize } f = c\sum_{i} P_i + \overline{T_N} \text{ with the } Q(i,k,j) \tag{30}$$

The minimisation can be achieved by selecting appropriate route control parameters $Q(x,m,y)$. Since we are interested in gradual optimisation in the presence of ongoing flows, we compute the partial derivative of f :

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

where we use the average power consumption from (27), (28):

$$\sum_{i\in\mathbf{N}} \frac{\partial P_i}{\partial Q(x,m,y)} = \sum_{k\in\mathbf{U}}[\sum_{i\in\mathbf{L}} \frac{\partial g_L(\Lambda_i)}{\partial(\Lambda_i)} \frac{\partial q(i,k)}{\partial Q(x,m,y)}\mu_i$$
$$+\sum_{i\in\mathbf{R}} \frac{\partial g_R(\Lambda_i)}{\partial(\Lambda_i)} \frac{\partial q(i,k)}{\partial Q(x,m,y)}(\mu_i + \Lambda^-(i,(i,k)))] \tag{32}$$

and the average delay from (29):

$$\frac{\partial \overline{T_N}}{\partial Q(x,m,y)} =$$
$$\sum_{k\in\mathbf{U}} \frac{\lambda^+(k)}{\Lambda_T^+}[\sum_{r\in\mathbf{R}} \frac{\pi(r,k)}{\Lambda_R(r,k)(1-q^2(r,k))} \frac{\partial q(r,k)}{\partial Q(x,m,y)}$$
$$+\sum_{l\in\mathbf{L}} \frac{\pi(l,k)}{\Lambda_L(l,k)(1-B^2(l))} \sum_{i\in\mathbf{U}} \frac{\partial q(l,i)}{\partial Q(x,m,y)}] \tag{33}$$

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 the cases where $h(x,y) = 1$, since when $h(x,y) = 0$ the partial derivatives will be 0. Now define the vector $\mathbf{q}_k = (q(1,k), q(2,k),..,q(N,k))$ and the $N \times N$ matrices $\mathbf{A}_k = [A_k(l,r)], \mathbf{D}_k = [D_k(r,l)], \mathbf{B}_k = [B_k(l,r)], \mathbf{C}_k = [C_k(r,l)]$:

$$A_k(l,r) = \frac{P(l,k,r)}{\mu_r + \Lambda^-(r,(r,k))} \tag{34}$$
$$D_k(r,l) = [P(r,k,l)\mu_r + \Lambda^-(r,(r,k))Q(r,k,l)) \tag{35}$$
$$B_k(l,r) = \mu_l P(l,k,r) \tag{36}$$
$$C_k(r,l) = \frac{[P(r,k,l)\mu_r + \Lambda^-(r,(r,k))Q(r,k,l)}{\mu_l[\mu_r + \Lambda^-(r,(r,k))]} \tag{37}$$
$$\tag{38}$$

and the $1 \times N$ row vectors :

$$\mathbf{N}(l) = 1/\mu_l \tag{39}$$

$$\mathbf{H}_k^{xmy}(l) = \begin{cases} \Lambda^-(x,(x,k))q(x,k) & k = m, l = y \\ -h(x,l)\Lambda^-(x,(x,k))q(x,k) & k = m, l \neq y \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

Then after some calculations we get:

$$\frac{\partial \mathbf{q}_k(r)}{\partial Q(x,m,y)} = \frac{\partial \mathbf{q}_k(r)}{\partial Q(x,m,y)}\mathbf{D}_k\mathbf{A}_k + \mathbf{H}_k^{xmy}\mathbf{A}_k \tag{41}$$

$$\frac{\partial \mathbf{q}_k(l)}{\partial Q(x,m,y)} = \frac{\partial \mathbf{q}_k(l)}{\partial Q(x,m,y)}\mathbf{B}_k\mathbf{C}_k + \mathbf{H}_k^{xmy}\mathbf{N} \tag{42}$$

where $r \in \mathbf{R}$ and $l \in \mathbf{L}$.

Thus, equations (41) and (42) can be written as:

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

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

$$W_k(i,j) = \begin{cases} \sum_{l\in\mathbf{L}} D_k(i,l)A_k(l,j) & i,j \in \mathbf{R} \\ \sum_{r\in\mathbf{R}} B_k(i,r)C_k(r,j) & i,j \in \mathbf{L} \end{cases} \tag{44}$$

$$\gamma_k^{xmy}(n) = \begin{cases} \sum_{l\in\mathbf{L}} H_k^{xmy}(l)A_k(l,n) & n \in \mathbf{R} \\ H_k^{xmy}(n)N(n) & n \in \mathbf{L} \end{cases} \tag{45}$$

So,

$$\frac{\partial \mathbf{q}_k}{\partial Q(x,y,m)} = \boldsymbol{\gamma}_k^{xmy}(\mathbf{I} - \mathbf{W}_k)^{-1} \tag{46}$$

where $\mathbf{I}$ is the $N \times N$ identity matrix. Using (46) we can calculate $\partial f/\partial Q(x,m,y)$ from equations (31), (32) and (33), and the matrix inversion is of time complexity $O(N^3)$. The corresponding 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 then determined by its n^{th} computational step:

$$Q_{n+1}(i,k,j) = Q_n(i,k,j) - \eta\frac{\partial f}{\partial Q(i,k,j)}|_{Q(i,k,j)=Q_n(i,k,j)} \tag{47}$$

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. First initialise all the values $Q(i,k,j)$ and choose $\eta > 0$.
2. Solve the system of the $U \times N$ nonlinear equations (5)-(12) based on the current state to obtain the $q(i,k)$.
3. Solve the system of $U \times N$ linear equations (46) using the $q(i,k)$.
4. Using the results from steps 3 and 4 update the values $Q(i,k,j)$ using (47)

2.2. Optimisation Through Network Balancing

Although it is of polynomial complexity, the above algorithm would be quite slow for a large network.

Thus we can now propose a much simpler cost minimising algorithm that also provides a gradient descent for the cost:

$$C = \sum_{r\in\mathbf{R}} \frac{\Lambda_r}{\Lambda_T}F_r(\Lambda_r) + \sum_{l\in\mathbf{L}} \frac{\Lambda_l}{\Lambda_T}F_l(\Lambda_l)$$

where Λ_r, Λ_l are the traffic rates at the routers and links, and Λ_T is the total traffic carried by the network.

If we incorporate the link drivers to the routers, this becomes:

$$C = \sum_{r \in \mathbf{R}} \frac{\Lambda_r}{\Lambda_T} F_r(\Lambda_r) \tag{48}$$

incorporating both delay and energy costs. The coefficient $\frac{\Lambda_r}{\Lambda_T}$ reflects the relative importance of the r^{th} router in terms of the proportion of traffic it carries. For the QoS component of the cost function this coefficient is justified, but we incorporate into the term $F_r(\Lambda_r)$ both the QoS term and energy consumption, and for the latter such a coefficient is justified if packets are being "charged a sum of money per packet" for the energy, while in the overall energy cost of the network such a coefficient may not be justified. However this is not essential, because starting with the actual energy consumption of the r^{th} node in watts $P_r(\Lambda_r)$, we can divide it by $\frac{\Lambda_r}{\Lambda_T}$ to obtain:

$$F_r^P(\Lambda_r) = \frac{\Lambda_T}{\Lambda_r} P_r(\Lambda_r), \tag{49}$$

and with the QoS cost $F_r^{QoS}(\Lambda_r)$ it yields the total router metric:

$$F_r(\Lambda_r) = F_r^{QoS}(\Lambda_r) + F_r^P(\Lambda_r) \tag{50}$$

For two distinct paths in the network of lengths u and v, with $v \leq u$, each composed of a sequence of nodes and links. The first path is $r_1, \ ... \ r_u, \ l_1, \ ... \ l_u$, while the second path is $r'_1, \ ... \ r'_v, \ l'_1, \ ... \ l'_v$. We will say that these paths are *balanced* if there exist traffic rates x_i, $1 \leq i \leq u$ such that:

$$F_r(x_i) = F_{r'_i}(x_i), 1 \leq i \leq v$$

Transferring traffic to achieve balanced paths, is an "ideal outcome" that may be impossible to attain.

However the condition of "balance" is indeed a desirable goal to achieve, because *we will now show that by balancing paths, the overall operating cost C associated with the network is reduced.*

Consider the continuous, increasing and differentiable functions $G_1(y)$, $G_2(y)$ of a real variable y, and a number $-\infty < x < +\infty$ for which $G_1(x) = G_2(x)$, with $G_1(y) \leq G_2(y)$ for $y \leq x$. Construct $G(y)$ such that:

$$\begin{aligned} G(y) &= G_1(y), \ y \leq x \qquad (51) \\ &= G_2(y), \ y \geq x \qquad (52) \end{aligned}$$

Then for small enough $\lambda > 0$,

$$(x+\lambda)[G(x) - G(x+\lambda)] + (x-\lambda)[G(x) - G(x-\lambda)] \leq 0$$

or

$$(x+\lambda)G(x+\lambda) + (x-\lambda)G(x-\lambda) \geq 2xG(x) \tag{53}$$

To prove this, note that since G_1 and G_2 are continuous and differentiable, so is $G(x)$, and for small enough $\lambda > 0$ we can use the Taylor series expansion to write $G(x+\lambda) \cong G(x) + \lambda G'(x)$, and $G(x-\lambda) \cong G(x) - \lambda G'(x)$ where $G'(x)$ denotes the derivative of $G(x)$, so that:

$$\begin{aligned} &(x+\lambda)[G(x) - G(x+\lambda)] + (x-\lambda)[G(x) - G(x-\lambda)] \\ &\cong -(x+\lambda)\lambda G'(x) + (x-\lambda)\lambda G'(x) \\ &\cong -2\lambda G'(x) \leq 0 \end{aligned}$$

because $G'(x) \geq 0$ since $G(x)$ is an increasing function of x. This leads to:

Theorem Let the F_r, $r \in \mathbf{R}$, be continuous, differentiable and increasing functions of their respective traffic rates Λ_r, Λ_l. If traffic balance can be achieved for nodes $r_1, \ ... \ r_u$ and $r'_1, \ ... \ r'_v$ with $v \leq u$ by shifting an amount of traffic $\lambda > 0$ from the former to the latter path, so that after the shift is carried out we have

$$F_r(\Lambda_{r_i} - \lambda) = F_{r'_i}(\Lambda_{r'_i} + \lambda), 1 \leq i \leq v,$$

with $\Lambda_{r_i} - \lambda = \Lambda_{r'_i} + \lambda$, then the resulting operating cost C' *after* this shift is *no greater* than the operating cost C *before* the shift, provided λ is small enough.

Proof Start with a network with given traffic rates, and then shift an amount of traffic λ so that nodes $r_1, \ ... \ , r_u$ carry less traffic, while nodes $r'_1, \ ... \ , r'_v$, $v \leq u$, carry more, leading to the new value of the cost function:

$$\begin{aligned} C' = C &+ \sum_{i=1}^{u} \frac{\Lambda_{r_i} - \lambda}{\Lambda_T} F_{r_i}(\Lambda_{r_i} - \lambda) \\ &+ \sum_{i=1}^{v} \frac{\Lambda_{r'_i} + \lambda}{\Lambda_T} F_{r'_i}(\Lambda_{r'_i} + \lambda) - \sum_{i=1}^{u} \frac{\Lambda_{r_i}}{\Lambda_T} F_{r_i}(\Lambda_{r_i}) \\ &- \sum_{i=1}^{v} \frac{\Lambda_{r'_i}}{\Lambda_T} F_{r'_i}(\Lambda_{r_i}) \end{aligned} \tag{54}$$

Multiply both sides by Λ_T and group terms to obtain:

$$\begin{aligned} &(C' - C)\Lambda_T + \sum_{i=1}^{u} \Lambda_{r_i} F_{r_i}(\Lambda_{r_i}) \\ &+ \sum_{i=1}^{v} \Lambda_{r'_i} F_{r'_i}(\Lambda_{r_i}) - \sum_{i=v+1}^{u} (\Lambda_{r_i} - \lambda) F_{r_i}(\Lambda_{r_i} - \lambda) \\ &= \sum_{i=1}^{v} [(\Lambda_{r_i} - \lambda) F_{r_i}(\Lambda_{r_i} - \lambda) + (\Lambda_{r'_i} + \lambda) F_{r'_i}(\Lambda_{r'_i} + \lambda)] \end{aligned} \tag{55}$$

The target balanced traffic rates for the routers are $x_i = \Lambda_{r_i} - \lambda = \Lambda_{r'_i} + \lambda$ for $1 \leq i \leq v$ while $x_i = \Lambda_{r_i} - \lambda$ for $v+1 \leq i \leq u$, so that we can write:

$$\begin{aligned} &(C' - C)\Lambda_T + \sum_{i=v+1}^{u} [(x_i + \lambda) F_{r_i}(x_i + \lambda) - x_i F_{r_i}(x_i)] \\ &+ \sum_{i=1}^{v} [(x_i + \lambda) F_{r_i}(x_i + \lambda) + (x_i - \lambda) F_{r'_i}(x_i - \lambda)] \\ &= \sum_{i=1}^{v} [x_i F_{r_i}(x_i) + x_i F_{r'_i}(x_i)] \end{aligned} \tag{56}$$

and define for $1 \leq i \leq v$:

$$\begin{aligned} G_i(x) &= F_{r_i}(x), \ x \geq x_i \\ G_i(x) &= F_{r'_i}(x), \ x \leq x_i \end{aligned} \tag{57}$$

so that we can write:

$$
\begin{aligned}
&(C' - C)\Lambda_T + \sum_{i=v+1}^{u} [(x_i + \lambda)F_{r_i}(x_i + \lambda) - x_i F_{r_i}(x_i)] \\
&+ \sum_{i=1}^{v} [(x_i + \lambda)G_i(x_i + \lambda) + (x_i - \lambda)G_i(x_i - \lambda)] \\
&= 2\sum_{i=1}^{v} x_i G_i(x_i) \qquad (58)
\end{aligned}
$$

Using (53) we have

$$(C' - C)\Lambda_T + \sum_{i=v+1}^{u} [(x_i + \lambda)F_{r_i}(x_i + \lambda) - x_i F_{r_i}(x_i)] \le 0$$

or

$$C' \le C - \frac{1}{\Lambda_T} \sum_{i=v+1}^{u} [(x_i + \lambda)F_{r_i}(x_i + \lambda) - x_i F_{r_i}(x_i)]$$

and because the right-most term is negative we have $C' < C$ completing the proof.

3. CONCLUSIONS AND FURTHER WORK

There is increasing interest in reducing energy consumption for all areas of human activity. Many approaches to energy efficiency and "green" systems, such as the Smart Grid, distance learning and E-work, in turn have a significantly higher dependency on computer networks and information technology; as a result they may reduce energy consumption in transportation (for instance), but in turn they increase the need for energy to run networks and ICT systems. Thus, as we seek more energy efficient ways to run our daily lives, we also need to seek more energy efficient ways to run ICT systems. Energy savings in new computing paradigms such as Cloud Computing can help, but in turn they can require more energy for operating the networks that will carry the data, the program code, and the results of the computations [2]. Thus some of the previous work of our group has addressed practical techniques for energy savings in packet networks based on re-routing packets and turning routers on and off depending on traffic conditions and on individual energy consumption of routers.

Therefore in this paper we have specifically addressed the question of how to use routing control in a network as a means to reduce energy consumption while remaining aware of QoS considerations, and proposed approaches based on network analysis and modelling that can be used to distribute traffic in the network so as to reduce an overall cost function that includes energy and QoS.

Two methods have been discussed. The first uses G-Networks to incorporate both the conveyance of user traffic and of traffic used to take control actions concerning packet re-routing. Because energy usage in routers and network links have different characteristics, and because links may be easier to turn off in order to save energy, while routers can be kept and can use less energy when they carry less traffic (due to the incorporation of multiple processor chips), we have developed a model that incorporates routers and links as separate queuing servers. This model has then allowed us to develop a gradient based algorithm for progressive traffic re-routing that runs in polynomial time complexity as a function of the size of the network. The second method we suggest is based on a result we have obtained that implies that load balancing actually will reduce both energy consumption and also improve QoS. This approach would be much simpler to implement but should be carefully considered so that opposing decisions, leading to oscillations, are not taken in the network.

We expect that future work will attempt to integrate both the experimental implementation of energy aware routing and the modelling and analysis techniques developed in this paper, so as to provide guidelines and practical means for both QoS and energy optimisation in networks.

4. ACKNOWLEDGMENT

The authors acknowledge the support for this research from the Fit4Green European Union FP7 Project co-funded under ICT Theme: FP7-ICT-2009-4.

REFERENCES

[1] Gelenbe, E. and Silvestri, S. (2009) Reducing power consumption in wired networks. *24th International Symp. on Computer and Information Sciences (ISCIS'09)*, METU, North Cyprus Campus, 14-16 September. (IEEE Digital Library).

[2] Berl, A., Gelenbe, E., Girolamo, M. D., Giuliani, G., Meer, H. D., Dang, M. Q., and Pentikousis, K. (2010) Energy-efficient cloud computing. *The Computer Journal, Advance Access Online*, **53**.

[3] Gupta, M. and Singh, S. (2003) Greening of the internet. *Computer Communication Review*, **33**, 19–26.

[4] Gupta, M. and Singh, S. (2007) Dynamic ethernet link shutdown for energy conservation on ethernet links. *Communications, 2007. ICC '07. IEEE International Conference on*, June, pp. 6156 –6161.

[5] Chabarek, J., Sommers, J., Barford, P., Estan, C., Tsiang, D., and Wright, S. (2008) Power awareness in network design and routing. *INFOCOM 2008. The 27th Conf. Computer Comms. IEEE*, April, pp. 457 – 465.

[6] Vasic, N. and Kostic, D. (2008) Energy-aware Traffic Engineering. Technical report. EPFL.

[7] Nedevschi, S., Popa, L., Iannaccone, G., Ratnasamy, S., and Wetherall, D. (2008) Reducing network energy consumption via sleeping and rate-adaptation. *NSDI'08: Proc. 5th USENIX Symposium on Networked Systems Design and Implementation*, Berkeley, CA, USA, pp. 323–336. USENIX Association.

[8] Chiaraviglio, L., Mellia, M., and Neri, F. (2009) Reducing power consumption in backbone networks. *Communications, 2009. ICC '09. IEEE International Conference on*, June, pp. 1 –6.

[9] Chiaraviglio, L., Mellia, M., and Neri, F. (2009) Energy-aware backbone networks: A case study. *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, June, pp. 1 –5.

[10] Chiaraviglio, L., Ciullo, D., Leonardi, E., and Mellia, M. (2009) How much can the internet be greened? *GLOBECOM Workshops, 2009 IEEE*, 30 2009-Dec. 4, pp. 1 –6.

[11] Gelenbe, E. (2004). Cognitive packet network. U.S. Patent 6,804,201.

[12] Gelenbe, E. (2009) Steps towards self-aware networks. *Communications of the ACM*, **52**, 66–75.

[13] Rajaraman, R. (2002) Topology control and routing in ad hoc networks: a survey. *SIGACT News*, **33**, 60–73.

[14] Santi, P. (2005) Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, **37**, 164–194.

[15] Jia, X., Li, D., and Du, D. (2004) Qos topology control in ad hoc wireless networks. *INFOCOM 2004. 23rd Annual Joint Conf. IEEE Computer and Comms. Societies*, March, pp. 1264 – 1272 vol.2.

[16] Boukerche, A., Cheng, X., and Linus, J. (2005) A performance evaluation of a novel energy-aware data-centric routing algorithm in wireless sensor networks. *Wireless Networks*, **11**, 619–635.

[17] Xu, Y., Heidemann, J., and Estrin, D. (2000) Adaptive energy-conserving routing for multihop ad hoc networks. Research Report 527. USC/Information Sciences Institute.

[18] Gelenbe, E. and Lent, R. (2004) Power-aware ad hoc cognitive packet networks. *Ad Hoc Networks*, **2**, 205–216.

[19] Lent, R. (March 2010) Power measurements of processors for routing. Technical report. ISN Group, EEE Dept., Imperial College, London, UK.

[20] Gelenbe, E. (1994) G-networks: An unifying model for queuing networks and neural networks. *Annals of Operations Research*, **48**, 433–461.

[21] Gelenbe, E. (1993) G-networks with signals and batch removal. *Probability in the Engineering and Informational Sciences*, **7**, 335–342.

[22] Fourneau, J.-M. and Gelenbe, E. (2004) Flow equivalence and stochastic equivalence in g-networks. *Computational Management Science*, **1**, 179–192.

[23] Gelenbe, E. (1993) G-networks with instantaneous customer movement. *Journal of Applied Probability*, **30**, 742–748.

[24] Fourneau, J.-M., Gelenbe, E., and Suros, R. (1996) G-networks with multiple classes of negative and positive customers. *Theoretical Computer Science*, **155**, 141–156.

[25] Gelenbe, E. and Labed, A. (1998) G-networks with multiple classes of signals and positive customers. *European Journal of Operations Research*, **108**, 293–305.

[26] Gelenbe, E. and Mitrani, I. (2010) *Analysis and Synthesis of Computer Systems*. Imperial College Press, World Scientific.

[27] Gelenbe, E. and Pujolle, G. (1998) *Introduction to Queueing Networks*, 2nd edition. John Wiley & Sons Ltd.

[28] Baliga, J., Hinton, K., and Tucker, R. S. (2007) Energy consumption of the internet. *Optical Internet, 2007 and the 2007 32nd Australian Conference on Optical Fibre Technology. COIN-ACOFT 2007. Joint International Conference on*, June, pp. 1 –3.

[29] Alonso, M., Coll, S., Santonja, V., Martnez, J.-M., Lpez, P., and Duato, J. (2007) Power-aware fat-tree networks using on/off links. *3rd International Conference on High Performance Computing and Communications (HPCC 2007)*, Houston, TX, pp. 472–483.

[30] Baliga, J., Ayre, R., Hinton, K., Sorin, W. V., and Tucker, R. S. (2009) Energy consumption in optical ip networks. *Lightwave Technology, Journal of*, **27**, 2391 –2403.

[31] Chase, J. S., Anderson, D. C., Thakar, P. N., Vahdat, A. M., and Doyle, R. P. (2001) Managing energy and server resources in hosting centers. *SOSP'01: Proceedings of 18th ACM Symposium on Operating systems principles*, October, pp. 103–116. ACM.

[32] Chen, G., He, W., Liu, J., Nath, S., Rigas, L., Xiao, L., and Zhao, F. (2008) Energy-aware server provisioning and load dispatching for connection-intensive internet services. *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, April, pp. 337–350. USENIX Association.

[33] Economou, D., Rivoire, S., Kozyrakis, C., and Ranganathan, P. (2006) Full-system power analysis and modeling for server environments. *Workshop on Modeling, Benchmarking and Simulation (MoBS)*, June.

[34] Fan, X., Weber, W.-D., and Barroso, L. A. (2007) Power provisioning for a warehousesized computer. *ISCA 07: Proceedings of the 34th annual international symposium on Computer architecture*, June, pp. 13–23. ACM.

[35] Gelenbe, E. and Muntz, R. R. (1976) Probabilistic models of computer systems - part i. *Acta Informatica*, **7**, 35–60.

[36] Panarello, C., Lombardo, A., Schembra, G., Chiaraviglio, L., and Mellia, M. (2010) Energy saving and network performance: a trade-off approach. *e-Energy 2010 - First International Conference on Energy-Efficient Computing and Networking*, April.

[37] Restrepo, J., Gruber, C., and Machuca, C. (2009) Energy profile aware routing. *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, june, pp. 1 –5.

[38] Soteriou, V. and Peh, L.-S. (2004) Design-space exploration of power-aware on/off interconnection networks. *IEEE International Conf. on Computer Design*, San Jose, CA, pp. 510–517. Ieee Computer Soc. 21 LOS ALAMITOS BBF28.

[39] Urgaonkar, B., Shenoy, P., Chandra, A., Goyal, P., and Wood, T. (2008) Agile dynamic provisioning of multi-tier internet applications. *ACM Trans. on Autonomous and Adaptive Systems (TAAS)*, **3**, 1–39.