GreeDi: An Energy Efficient Routing Algorithm for Big Data on Cloud

T. Baker, B. Al-Dawsari

School of Computing and Mathematical Science, Liverpool John Moores University, UK
t.baker@ljmu.ac.uk, b.m.aldawsari@2012.ljmu.ac.uk

H. Tawfik, D. Reid

Department of Mathematics and Computer Science, Liverpool Hope University, UK
tawfikh@hope.ac.uk, reidd@hope.ac.uk

Y. Ngoko

Laboratoire dInformatique de Paris Nord-Institut Gallillee, France
yanik.ngoko@lipn.univ-paris13.fr

Abstract

The ever-increasing density in cloud computing parties, i.e. users, services, providers and data centres, has led to a significant exponential growth in: data produced and transferred among the cloud computing parties; network traffic; and the energy consumed by the cloud computing massive infrastructure, which is required to respond quickly and effectively to users requests. Transferring big data volume among the aforementioned parties requires a high bandwidth connection, which consumes larger amounts of energy than just processing and storing big data on cloud data centres, and hence producing high carbon dioxide emissions. This power consumption is highly significant when transferring big data into a data centre located relatively far from the users geographical location. Thus, it became high-necessity to locate the lowest energy consumption route between the user and the designated data centre, while making sure the users requirements, e.g. response time, are met.

The main contribution of this paper is GreeDi, a network-based routing algorithm to find the most energy efficient path to the cloud data centre for processing and storing big data. The algorithm is, first, formalised by the situation calculus. The linear, goal and dynamic programming approaches used to model the algorithm. The algorithm is then evaluated against the baseline shortest path algorithm with minimum number of nodes traversed, using a real Italian ISP physical network topology.

Keywords: Big Data; Cloud Computing; Routing Algorithm; Data Centre

1. Introduction

During the last decade, the use of cloud computing to run businesses and individual based services has increased rapidly based on an *on − demand* pay-as-you-need pattern. This is due to the very simple cloud computing services provision model: providers offer high performance computing resources to end users; end-users subscribe to the resources they need, and obviously a high-speed network connection must be established between users and providers to formulate the model. The International Data Corporation (IDC) published in [1] surprising figures, which showed that the global cloud computing services use (i.e. network-based storage) increased from $16 billion in 2008 into $42 billion in 2012; which inevitably required that cloud computing provides strong storage, computation and distributed capability to structure and process the big data (e.g. medical records, video and image archives, scientific applications) produced by all the above cloud computing parties.

This enormous growth in cloud services, display and demand, is expected to generate revenues of nearly 35 billion euro just in Europe by 2014 [2]. The expectation was the spark for the biggest companies in the world (e.g. Google, Amazon, Cisco) to start heavily investing in cloud computing infrastructure and data centres. Not only big companies were aimed to build their own data centres, but also other enterprises and institutions (e.g. academic institutions) are all now planning to have their own private and public cloud data centres. For example, University of Salford Manchester made £5.7 million in cloud computing investment [3].

This ever-increasing density in cloud computing users, providers, and data centres have led to significant increases in network traffic and the associated energy consumed by the huge infrastructure (e.g. extra servers, switches) required to respond quickly and effectively to users requests. Moreover, transporting data between data centres and cloud users can consume even larger amounts of energy than just processing and storing the data on the cloud data centres [4], and hence producing high carbon dioxide emissions. This power consumption is particularly significant when transferring data into a data centre located somewhere in the world relatively far from the user geographical location; for example, the user is based in Liverpool in the UK and Google data centre is in Hong Kong [5]. In addition, the higher bandwidth and high speed network required to cope with the cloud network traffic and to speed up data transformation process generates higher carbon dioxide footprint [6]. This is against the environmental requirements published by the 2011 report of PBL Netherlands Environmental Assessment Agency and JRC European Commission [7] and also in [8] to reduce the energy consumption to decrease the CO_2 emission volume by 15%-30% before 2020 to keep up the global temperature increase below 2^oC. Thus, the rapidly growing energy consumption and CO_2

emission of using the cloud computing has become a key environmental concern.

Energy efficient routing solution for cloud computing is required to ensure the environmental sustainability. Since the data centres energy consumption has seen great deal of interest and work in the last years, however, cloud computing network energy consumption is still in its infancy and requires further research and development to be fully achieved.

The rest of the paper is organised as follows: the next section highlights the main aim and objectives of the paper. The related works have been summarised in section 3; the proposed model will be discussed in section 4, which includes discussion about the situation calculus use to analyse the network topology. Linear, Goal and Dynamic programmingmodelling approaches will be discussed in section 5. The evaluation of the proposed model is detailed in section 6; finally, the paper concludes the results and paves the future work in section 7.

2. Aim and Objectives

There are two main pillars for energy consumed at cloud computing that should be dealt with efficiently and equally to achieve the full green cloud computing network:

(i) the amount of energy consumed at the data centre and

(ii) the amount of energy consumed on transporting the data between the user and the cloud data centre.

Since the current state-of-the-art solutions focus primarily on improving the energy consumed at the data centres, as the next section shows; thereby, the primary aim of this paper is to propose and evaluate a high-end routing algorithm entitled Green Director (GreeDi) to address the gap. GreeDi acts as an intermediary bridge for directing the users requests to the green data centres based primarily on using the most energy efficient route to achieve the full green cloud computing network ambition while making sure the users requirements, e.g. response time, are met.

To accomplish this aim, we first model the cloud computing network and its power consumption as a basis to compute the energy required by the cloud network before and after using the proposed algorithm. We will then formalise the interconnection between the cloud user and a green data centre, by using a situation calculus model to define the logical state of the network. Once the interconnection is established and formalised, we then calculate the energy required for the transportation. Linear programming approach will be used thereafter to model the proposed algorithm, which will finally be evaluated against the well-known shortest path routing policy.

3. Related Work

Energy consumption and analysis has been studied from different computer science domains and perspectives, such as from hardware architecture, software architecture network or even I/O technologies (i.e. storage). A discussion of the various elements that contribute to the total energy consumption in cloud environments and how they are addressed in the literature can be found in [9]. However, the most relevant solutions and the very interrelated ones to this work will be discussed in this section. Many energy efficient routing algorithms and protocols have been proposed in the literature, such as [10, 11, 12], where they can be classified into two main categories: (I) Minimum Energy routing algorithms [10] and (II) Maximizing Network Lifetime routing algorithms [11, 12]. The first one tries to find the most energy efficient route to transmit the data packets from sender to receiver; whereas the second one aims at balancing the remaining battery power in each of the intermediate nodes.

In turn, The Minimum Energy routing algorithms can be divided into three sub-classes based on the types of link costs:

(i) Minimum Total Transmission Power (MTTP) [13], which uses the transmission power as the link metric and search for the path with minimum total transmission power between the sender and the receiver. For example, the authors in [13] modified the Dijkstras Shortest path algorithm to get the MTTP path.

(ii) Minimum Total TransCeiving Power (MTTCP) [14], which uses the transmission power as well as the receiving power as the link cost.

(iii) Minimum Total Reliable Transmission Power (MTRTP) [12], which uses the total transmission power for transmitting the data packets from one node to its adjacent node reliably as the link cost.

However, the above algorithms did not consider the energy and time required for computing/processing a job (receive, process, forward) at each individual intermediate node, which is critical in moving big data to cloud data centre. Recent studies in [15] and [16] presented two online algorithms to solve the issues of moving big data to the cloud. First, Online Lazy Migration (OLM) Algorithm that seeks to prevent moving a large amount of data back and forth too often by avoiding aggressive switches of the aggregation data centre. The second is a Randomized Fixed Horizon Control (RFHC) algorithm, which can predict and exploit future information based on Markov chain model. Based on their evaluation, (OLM) and (RFHC) can achieve a very low competitive ratio; however, they concluded that energy efficient routing solution for cloud computing is essential to ensure environmental sustainability.

The authors in [17] presented a new routing strategy to reduce the cloud network CO_2 emissions by dynamically routing/transferring the on-demand energy-intensive data processing requests, via IP-over-WDM networks, to data centres that are powered primarily by renewable energy sources such as wind and solar. However, this solution helps in reducing the CO_2 emissions at data centres level only, and not at the data transportation level.

Another complementary research reported in [18] studied the energy consumption in both: the data centre and in data transportation to data centres. They have used optical networks and virtualisation in IP-over-WDM architecture to save the power in the data centres and achieve green communication. Two models are proposed in that research:

(i) Delay-Minimized Provisioning (DeMiP), which aims to select the nearest data centre based on pre-computed distances between nodes in virtual topology, and then the virtual links from the virtual topology are mapped on physical topology by utilising Dijkstras algorithm for shortest path;

(ii) The Power-Minimized Provisioning (PoMiP), which focuses on IP routers as power consumers in the transport network and aim to minimise the utilisation of the IP router ports. It selects the virtual link with low-power.

Another study presented in [19] targeted the data centre level but from a different angle in which it tries to reduce the data centre power consumption while guarantee the service performance based on the users' perspective and expectation. Their proposed software architecture enables comprehensive online-monitoring, live virtual machine migration, and VM placement optimisation.

Other proposals such as [20, 21] try to understand and find a tradeoff between energy efficiency and performance. In particular, in [20], the authors study such a tradeoff using a social media analysis case study, where there is likely to be a high level of variability (both in performance and energy use). In [21], the tradeoff between power consumption and Service Level Agreement (SLA) enforcement is formulated as constraint satisfaction problems, and it is developed within the context of cloud computing data centres.

In [22], a framework that integrates energy awareness, and even environmental impact as a part of the SLA is proposed. The authors identified several parameters that could be used within a SLA, such as the amount of CO_2 correlated with environmental measurements that are easier to measure and understand for a user. An interesting study in [23] presents a cloud energy management system by using a sensor management function and a VM allocation tool. These sensors are deployed across multiple data centres and can be accessed and monitored via a unified interface for those multiple data centres. The collected

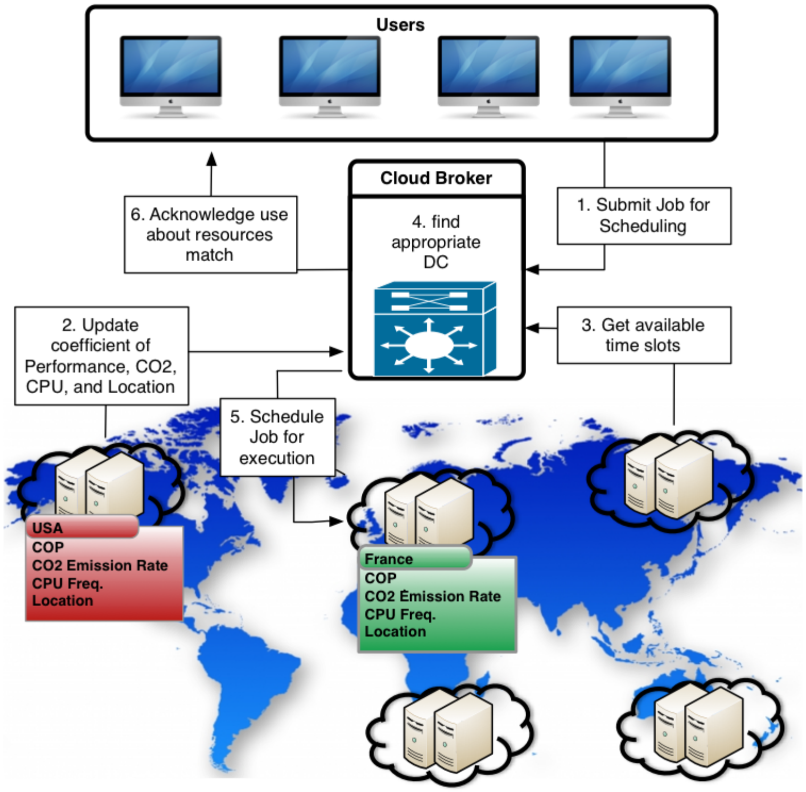

Figure 1: Cloud Broker Network Overview

data are used and analysed via the sensor management function through four main phases: Monitoring, Calculation, Analysis, and Action. The study achieved 30% energy reduction at data centres level.

Another interesting approach is the one presented in [24]. It proposes a number of algorithms to reduce power consumption by consolidating the hosting workloads and shutting down physical machines, which become idle after consolidation. Linear programming has been utilised for the scheduling of tasks within heterogeneous clusters [25]. By simulation, they showed that using their proposed policy results in significant reduction in energy consumption. In the domain of automated workflow compositions and cloud computing, an autonomic service composition engine based on AI techniques for cloud environments was proposed and described in [26]. The paper presented a formal approach that uses Situation Calculus to translate service requirements into an Intention Workflow Model (IWM). This IWM is then used to generate autonomic cloud service composition.

4. Proposed Model: Basics and Rules

Since the previous research were focused on how to achieve green data centres, this has helped us using the following assumption throughout the paper:

There are n *green data centres to which a user machine i can be connected to through the Internet, to accomplish a certain task.*

So, one of these available data centres will be used, by which it must be accessible via the selected most energy efficient route. In other words, amongst multiple routes to a green data centre, the most energy efficient route will be chosen by the new framework.

4.1. Modelling power consumption of the network

Modelling power consumption of the cloud network is an essential part of this work since it is the basics that the rest of the paper and other calculations and algorithms proposed here are all based on. According to [27, 28, 29, 30], one of the most widely accepted methods for modelling the power consumption of massively distributed network infrastructure, such as cloud network, is based on the telecommunications equipment inventory statistics and their historical sales figures (i.e. once the quantity and type of equipment in the network are known, the energy consumption of these equipment can be easily calculated). However, this approach alone cannot determine the actual network architecture and structure. Once the network architecture is know, then required components can be identified and energy consumption can be calculated accordingly.

As discussed in [31, 32], telecommunications network-based model is an essential approach to be used side-to-side with the above one to address the gap. In this approach, the network is partitioned into a number of main parts: access network, metro/edge network, core network, data centre and IPTV web services network. The network model presented in Figure 1 is a *first-cut* of such a massively distributed network, and as such it does not include much of the fine details of the network true structure and topology. However, it does show the main network architecture and the required components, which are needed for the energy consumption calculation purposes. The energy consumption of the network is calculated using the manufacturers data on equipment quantity and energy consumption for a range of typical types of equipment for each part of the network. Using a combination of the above two approaches help in calculating the power consumption of the entire network using real world network infrastructure components; and it also helps in predicting the growth in power consumption depends on the network architecture, and the equipment inventory statistics and their historical sales figures provided by the manufacturers.

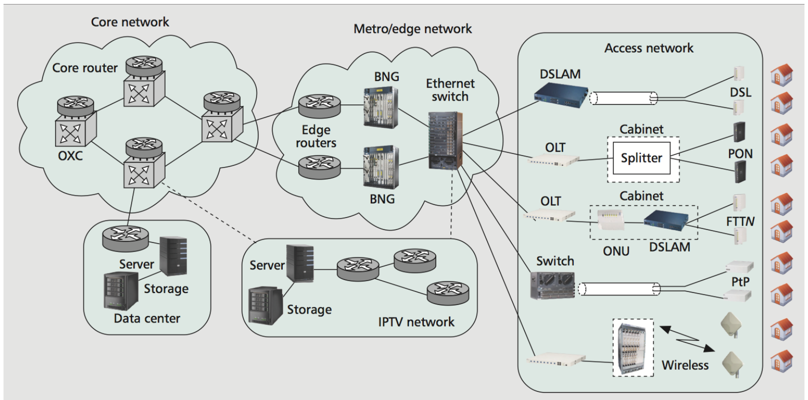

Figure 2: Distributed network structure [33]

4.2. Modelling user connectivity to data centre

The interconnection between a user machine i and a data centre DC_i, via the proposed algorithm, is based on the public cloud structure shown in Figure 1 above and also [34], which will, in this paper, be formalised as a graph. Thus, between any i and a DC_i, we assume that we have an interconnection graph $G^i = (V^i, T^i, P^i, C^i, E^i, L^i, B^i)$.

Where V^i gives a list of all possible nodes available between any i and a DC_i; and $T^i : V^i \longrightarrow \{1, \ldots, 6\}$ states the nodes' types, which are six different types of nodes available. Therefore, as shown in Figure 1, each node v, where $v \in V^i$, might be: an ethernet switch ($T(v) = 0$), a broadband gateway router ($T(v) = 1$), a data centre gateway router ($T(v) = 2$), a provider edge router ($T(v) = 3$), a core router ($T(v) = 4$), and a high-capacity Wavelength Division Multiplexed (WDM) transport equipments/links ($T(v) = 5$), to interconnect the core routers, part of the public Internet.

$P^i(v)$ and $C^i(v)$ states the power consumption and the capacity of a node $v \in V^i$, respectively.

$E^i \subseteq V^i \times V^i$ defines the interconnection nodes; $L^i : E^i \longrightarrow \mathbb{N}$ gives the latency between connected nodes E^i; and finally B^i denotes to the bandwidth.

4.3. Formal analysis of network topology

In initial analysis, this network is highly distributed and does not necessarily possess global knowledge of its own state. It is thus necessary to apply a formalism to define the logical state of the user's network that does not rely on explicit state enumeration. For this, a calculus of situations is proposed as in [35] whereby a situation is the history of previous actions and one situation, s, is transformed into another, $do(a, s)$, by

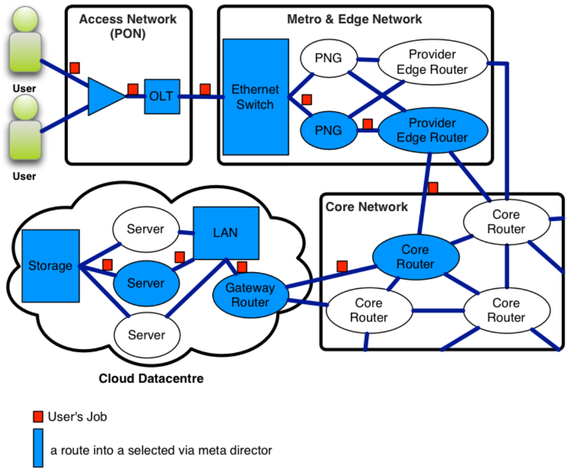

Figure 3: Network structure

applying action a in situation s. The logical state of the system is then determined from the initial conditions, effect axioms, frame axioms and qualification axioms. As discussed in [35], the frame problem can often be worked around combining frame and effect axioms into successor state axioms. Thus, the fitness of a node $v \in V^i$ in users i's network may be given by:

$fitness(v, do(a, s)) = F \Leftrightarrow [fitness(v, s) = F \wedge a \notin A(v)] \vee \exists m[fitness(v, s) = F - m \wedge value(a, s) = m]$

where $A(v)$ is the action set of node v and $value$ is a function from the set of actions to the integers, mapping each action to a reward (positive integer), cost (negative integer) or no effect (0). So, a relatively simple value system can be defined to optimise the choice of components based on discovering the user's most energy efficient data centre network, or indeed to match any user requirement. For example, adding a situation term to the capacity and power measures of a node:

$value(addComponent(v), s) = r \equiv P^i(v, s) > P^i(v, do(addComponent(v), s) \wedge r = 50 \vee$

$C^i(v, s) < C^i(v, do(addComponent(v), s) \wedge r = 100 \vee [(P^i(v, s) < P^i(v, do(addComponent(v), s)) \vee (C^i(v, s) > C^i(v, do(addComponent(v), s)) \wedge r = -10]$

Thus, for each node: $fitness(do(a, s)) = fitness(s) + value(a, s)$. Additionally, probabilities can be assigned to reflect the likelihood of success for network operations. Thus, it can be stated that for any node, the likelihood of connecting/routing to another node, $v\prime$ is dependent on the fitness of that node:

$$prob(addLink(v\prime), s) = \frac{f_{v\prime} d_{v\prime}}{\sum_{v \in V^i} f_v d_v} \tag{1}$$

Alternatively, for any $v \in V^i$ this is the probability that $(v, v\prime) \in E^i$. Thus, in order to maintain connectivity:

$connections(v, do(a, s)) = N \Leftrightarrow [(connections(v, s) = N) \wedge a \neq connection\_transfer] \vee connections(v, s) = N - m \wedge \exists v\prime(failed(v\prime, s) \wedge connections(v\prime, s) = m) poss(connection\_transfer, s) \Rightarrow \exists v\prime failed(v\prime, s)$

So, if v is the existing node that acquires the connections of a failed node, and $v\prime$ is the failed unit and $v\prime\prime$ is a new node created, then:

$connections(v, do(connection\_transfer, s)) = connections(v, s) + connections(v\prime, s) - m_{vv\prime\prime} - h_{vv\prime}$

$connections(v\prime\prime, do(connection\_transfer, s)) = 1 - - - [A]$

where $h_{vv\prime} = 1$ if v was connected to $v\prime$, and 0 otherwise, and $m_{vv\prime}$ is the number of nodes with mutual links to v and $v\prime$: $m_{vv\prime} = \sum_u h_{vu} h_{v\prime u}$

The probability that $(v, v\prime) \in E^i$ is

$$\frac{connections(v, s) connections(v\prime, s)}{nodeNumber(s) < connections(V^i, s) >}$$

where $< connections(V^i, s) >$ is the average number of connections in V^i. Denoting c_v for the number of connections a node v has in s, and N for the number of nodes in s, so the average number of connections to v and $v\prime$ is:

$$< m_{vv\prime} >= \sum_u \frac{c_v c_u}{N < c_{V^i} >} = K c_v c_{v\prime}$$

where $K = \dfrac{< c^2_{V^i} >}{N < c_{V^i} >^2}$

Now, substituting the expressions given in [A] in the probability generating function for a power law distribution

$$P(c) = \frac{1}{2} E(c^{c_{v\prime} - m_{vv\prime} - h_{vv\prime} - h_{v\prime v} + c_v}) + \frac{1}{2} c = \frac{1}{2}\Big[c + E(c^{c_v + c_{v\prime\prime}} e^{K c_v c_{v\prime} g(c)} (1 + \frac{c_v c_{v\prime} g(c)}{N < c_{V^i} >}))\Big]$$

where $g(c) = (1 - c)/c$ and $h_{vv\prime}$ is a random bit. As $N \to \infty$, K and $\dfrac{1}{N < c_{V^i} >} \to 0$.

So, letting $K^* = \dfrac{1}{N < c_{V^i} >}$. $P(c)$ can be expanded as a convergent power series in K and K^*. The first term when $K = K^* = 0$ gives $2P(c) = P^2(c) + c$ so that: $P(c) = 1 - \sqrt{1 - c} = \dfrac{1}{2\Gamma(\frac{1}{2})} \sum_{v=1}^{\infty} \dfrac{\Gamma(v - \frac{1}{2})}{v!} c^v$

Thus as $N \to \infty P(c_{V^i}) = \frac{\Gamma(c_{v^i} - \frac{1}{2})}{2\Gamma(\frac{1}{2})c_{v^i}! \sim c_{V^i}^{-3/2}}$.

Thus, the connectivity of the user's network to the data centre is governed by a power law distribution with exponent 1.5. Properties of such networks can then be utilised in identifying the most relevant green data centre to the user. For example, a neighbour node of any particular node is likely to have greater connectivity than that node [36, 37]. In this way, a strategy of ascertaining a green network overlay can be pursued based on identifying only key nodes in the network and routing over these.

4.4. Energy required for transportation

For any user's job to be processed, we assume that we have: the quantity of *Flops* that it requires w_u; the amount of input bits in_u to be processed; the amount of output bits ou_u to be returned.

Therefore, if we need an energy of $ET_{send}(i)$ for sending a bit from the user to the data centre and $ET_{recv}(i)$ for the inverse sending, the total energy transportation cost required for processing J_u is: $in_u.ET_{send}(i) + ou_u.ET_{recv}(i)$. To model $ET_{send}(i)$ and $ET_{recv}(i)$, we assume that data sent from a user machine to a data centre is always routed on a path that rely the two points connection *(the shortest path)*. In using the formulas proposed in [34], the energy required for sending one bit from a user to a data centre is:

$$ET_{Send}(i) = 6\left(\frac{3P^i_{es}}{C^i_{es}} + \frac{P^i_{bg}}{C^i_{bg}} + \frac{P^i_{g}}{C^i_{g}} + \frac{2P^i_{pe}}{C^i_{pe}} + \frac{18P^i_{c}}{C^i_{c}} + \frac{4P^i_{w}}{C^i_{w}}\right) \tag{2}$$

where in this case, $P^i_{es}, P^i_{bg}, P^i_{g}, P^i_{pe}, P^i_{c}$, and P^i_{w} represent the power consumed by the nodes types listed in subsection 4.2, Ethernet switches, broadband gateway routers, data centre gateway routers, provider edge routers, core routers, and WDM transport equipment, that are located on the path used for routing a user's job to a DC_i. C^i_{es}, C^i_{bg} , C^i_{g} , C^i_{pe}, C^i_{c} and C^i_{w} are the capacities of the corresponding equipment in bits per second. The values P^i and C^i depend on the nodes used.

Since the above equation doesn't take into account the power consumption of the other overheads in the cloud network, hence, the entire equation is multiplied by the left factor (six). The factor of six stands precisely for the power requirements for cloud redundancy (factor of 2), cooling equipments and other overheads (factor of 1.5), and the fact that todays network typically operates at under 50% utilization [38] while still consuming almost 100% of maximum power (factor of 2) [39]. The factor of three for Ethernet switches is to include the Ethernet switches in the metro network as well as the Ethernet switches in the LAN inside the data centre. The factor of two for provider edge routers is to include the edge router in the edge network and the gateway router in the data centre, and in the same vein for the other factors in the equation.

Let's consider that G^i comprises the set of paths $Pth = \{pth_1, \ldots pth_l\}$ from a user machine i to the data centre DC_i. Then, if the path pth_p was used for sending data, we will have $P^i_{es} = \sum_{(u,v)\in pth_p | T(v)=0} P^i(v)$ and $C^i_{es} = \sum_{(u,v)\in pth_p | T(v)=0} C^i(v)$.

And in the same vein, for the other nodes' types. For example, for the broadband gateway router, P^i_{es} and C^i_{es} will consecutively be: $\sum_{(u,v)\in pth_p | T(v)=1} P^i(v)$ and $\sum_{(u,v)\in pth_p | T(v)=1} C^i(v)$.

4.5. Time required for transportation

We assume a simple communication model, *Store and Forward*, where each node waits for a complete reception of the data before processing it. The approximate time required for sending α bits on a link $e \in E^i$ is equal to: $\max\{L^i(e), \lceil \frac{\alpha}{B^i(e)} \rceil . L^i(e)\}$.

where, as mentioned in subsection 4.2 above that, $L^i : E^i \longrightarrow \mathbb{N}$ gives the latency between connected nodes $e \in E^i$; and B^i denotes to the bandwidth. The idea behind it is that either, the bandwidth can contain the bits to send or, we must divide the data to send it in various blocks based on the bandwidth. Finally, we assume that the paths pth_p and $pth_{p'} \in Pth$ were used for sending user data in both directions; then, the total time required for the transportation of a Job J_u in both directions is equal to:

$$Tr(u,i) = \sum_{e\in pth_p} \max\{L^i(e), \lceil \frac{in_u}{B^i(e)} \rceil . L^i(e)\} + \sum_{e\in pth_{p'}} \max\{L^i(e), \lceil \frac{ou_u}{B^i(e)} \rceil . L^i(e)\} \tag{3}$$

4.6. Energy and time required for computation

We assume that each job J_u will be processed by a single machine in the data centre. We also assume that each data centre DC_i is made of a finite set of homogeneous machines that consume $EP(i)$ for processing one flop. Therefore, for processing a job J_u, the data centre DC_i will consume $w_u.EP(i)$.

Finally, any machine in a data centre DC_i needs approximatively $\mu(i)$ time units for processing one flop. The job J_u can then be processed in approximatively $w_u.\mu(i)$ times units.

5. Modelling Approach

5.1. Linear programming formulation

For processing users' jobs, the proposed GreeDi algorithm will be used, which routes users' jobs to the subscribed green data centres via the most energy efficient route; in order to minimize the energy consumption and the Service Response Time (SRT). However, this creates the following computational problem that must be solved: there are m users' jobs $J_1, \ldots, J_m$ that have been submitted to the framework gateway. This gateway can be seen here as a server machine connected to each data centre DC_i by an interconnection graph G^i. Each user's job is submitted with an *intention* file, which includes non-functional SLA requirements such as the maximal response time that the user expect for processing his request/job. Finally, each data centre DC_i is associated with a capacity q_i stating the maximal number of jobs that GreeDi algorithm can route on it. This parameter depends on negotiations, made between the framework and cloud provisioner. As we will see also, q_i is important for ensuring a minimal response time in the treatment of users' requests. The framework gateway must choose for each job J_u a data centre such that: minimise the total energy consumption (both transportation and processing as discussed earlier), while ensuring processing data within a minimal response time defined by the user's intention file. For a linear programming formulation, we consider a decisional variable $x(i, u) \in 0, 1$ which states whether or not the job J_u will be processed on the data centre DC_i. Let us denote to the Maximal Service Response Time, for processing job J_u (as defined in user's intention file), by $MSRT_u$. Thus, the problem can be represented as the following mixed integer linear programme:

Model LP_1 :

Minimize $Z = \sum_{u=1}^{m} \sum_{i=1}^{n} x(i, u).[w_u.EP(i) + in_u.ET_{Send}(i) + ou_u.ET_{Recv}(i)]$

Subject to:

1. $\forall J_u, DC_i : x(i, u) \in \{0, 1\}$
2. $\forall J_u : \sum_{i=1}^{n} x(i, u) = 1$
3. $\forall J_u : \sum_{i=1}^{n} x(i, u).[w_u.\mu(i) + Tr(u, i)] \leq MSRT_u$
4. $\forall DC_i : \sum_{u=1}^{m} x(i, u) \leq q_i$

Any LP_1 solution states to route job J_u towards the data centre DC_i if $x(i, u) = 1$. The constraint 3 in this modelling defines the maximal response time expected by users. Such a maximum can only be guaranteed if there is limitations on the maximal number of Jobs that can be processed in parallel in any data centre; hence constraint 4, in which q_i denotes to the maximal number of Jobs.

In LP_1, we assumed that for sending user data in both directions, two paths, pth_p and $pth_{p'}$, are used. Different paths selections might lead to different values of Z. For including this combinatory we have two options: (i) we include it in LP_1. It is hard in this case to avoid non linear equations; (ii) we execute multiple times the linear program with different path choices return the answer that leads to the minimisation of Z. This approach is more efficient since we remain with a linear model. Algorithm 1 below summarises LP_1.

Algorithm 1 LP1 Input, Output, Steps

INPUT: Jobs $J_1, \ldots, J_m$ with workloads, inputs and outputs data, and intention files; Data centres $DC_1, \ldots, DC_n$ with energy consumption per flop and frequency; Interconnection graphs $G^1, \ldots G^n$

OUPUT: Return the best solution on Z

STEPS:

1. Define, for each i, a set of paths $Cpth_i$ that can be used for sending and receiving data.
2. For each i, choose a pair of paths $(pth_p, pth_{p'}) \in Cpth_i$
3. Compute the resulting values of $ET_{Send}(i)$ and $ET_{Recv}(i)$ (equation 2);
4. For any job J_u and data centre DC_i compute $Tr(u, i)$ (equation 3)
5. Run LP_1 and obtain Z; if it is the best obtained value then it will be kept.
6. If there is possible combination $(pth_p, pth_{p'})$ that has not been explored, goto 2

In the special case where $Cpth_i$ is defined by taking the shortest paths on the bandwidth, we do not loop in this algorithm. It is trivial to observe that the intents set by users on maximal response time can make LP_1 unrealisable. To circumvent this difficulty, we will adopt a goal programming formulation.

5.2. *Goal programming formulation*

For any job J_u, we introduce two real deviation variables d_u^+ and d_u^-. A job J_u can be put on data centre DC_i if:

$$w_u.\mu(i) + Tr(u, i) + d_u^- - d_u^+ = MSRT_u$$

In order to approximate to user's intents, we must minimise d_u^+ (the excess between the effective SRT and the one expected by the user). Our objective within this new perspective is to minimise both: the deviation on user requirements and the total energy consumption. For handling these two separate objectives in a same function, we assume that there is a preference factor β_u defined on each job by the user and stating what is the relative importance of SRT minimisation over the energy consumption. From the above, we can derive the following model:

Model LP_2 :

$Minimize \sum_{u=1}^{m}(1-\beta_u)\frac{E_u}{E_u+d_u^+} + \beta_u.\frac{d_u^+}{E_u+d_u^+}$

Subject to:

1. $\forall J_u, DC_i : x(i,u) \in \{0,1\}$
2. $\forall J_u : \sum_{i=1}^{n} x(i,u) = 1$
3. $\forall J_u : d_u^-, d_u^+ \geq 0$
4. $\forall J_u : \sum_{i=1}^{n} x(i,u).[w_u.\mu(i) + Tr(u,i)] + d_u^- - d_u^+ = MSRT_u$
5. $E_u = \sum_{i=1}^{n} x(i,u).[w_u.EP(i) + in_u.ET_{Send}(i) + ou_u.ET_{Recv}(i)]$
6. $\forall DC_i : \sum_{u=1}^{m} x(i,u) \leq q_i$

The above modelling is based on two type of goals: users ones (submitted via intention which is a part of SLA) and the energy minimisation. In the current formulation, we reduced the users intents to a threshold for the SRT, as per constraint 4 of MSRT in the above LP2. But, it makes sense to envision an extension of the modelling to include other requirements such as a maximal price or a minimal level of security for its data. Finally, let us observe that we *normalised* the Energy and the goal deviation in order to make these quantities comparable. This choice has however a drawback: the objective function becomes non-linear. As such, we will propose a dynamic programming approach for computing fast solutions of LP_2.

5.3. Dynamic programming approach

In this solution we maintain a two dimensional array $Z \in \mathbf{R}^{n\times m}$. Each $Z(i,l)$ corresponds to an assignment of the jobs $J_1, \ldots J_{l-1}$ to data centres in which J_l is associated with the data centre DC_i. At the beginning of the algorithm, we compute

$$Z(i,1) = (1-\beta_1)\frac{E_1(i)}{E_1(i)+d_1^+(i)} + \beta_1.\frac{d_1^+(i)}{E_1(i)+d_1^+(i)}$$

for any data centre DC_i. Here,

$$E_1(i) = w_1.EP(i) + in_1.ET_{Send}(i) + ou_1.ET_{Recv}(i)$$

and

$$w_1.\mu(i) + Tr(1,i)] + d_1^-(i) - d_1^+(i) = MSRT_1$$

For the computation of $Z(i,l), l > 1$, we proceed as follows:

1. We consider the different assignments $Z(1, l-1), \dots Z(n, l-1)$ in which the number of Jobs assigned to DC_i is lower than q_i. We will refer to these assignment as (i, l) compatible ones.
2. If there are no (i, l) compatible assignments, we set $Z(i, l) = +\infty$
3. Otherwise, we choose the (i, l) compatible assignment with the smallest objective value and sum this value in $Z(i, l)$, with the cost required for assigning J_l to the data centre DC_i. In a formal manner, this cost is

$$Z^{+}(i, l) = (1-\beta_l)\frac{E_l(i)}{E_l(i) + d_l^{+}(i)} + \beta_l.\frac{d_l^{+}(i)}{E_l(i) + d_l^{+}(i)}$$

At the end, we have the values of $Z(i, n)$, computed for each data centre DC_i. We then select the assignment that leads to the smallest objective value.

The Bellman rule of this modelling can be resumed as follows: *the optimal assignment of Job J_l on the data centre DC_i is obtained from the optimal assignment of Jobs $J_1, \dots, J_{l-1}$ in which the capacity used for the data centre DC_i is lower than q_i*. The optimality of this rule can be influenced by the way we sort the jobs. We propose for this to use the user's submission ordering. That is: J_1 is the first submitted Job, J_2 the second etc. The advantage of this ordering is that implicitly, *the first user will have the best services*.

6. Evaluation

The scenario presented in this section is aimed to illustrate how the total energy transportation results on directing a user's request to a subscribed green data centre. In the following, we present first the network physical topology used for the energy efficiency evaluation, then we detail the nodes types per route in the presented topology and compute the energy consumption to compare the eventual results.

6.1. Physical topology

The network topology we used exploits the hierarchal design that is physically used by an Italian Internet Service Provider (ISP) [40]. Four levels of nodes presented in this topology, namely: core, backbone, metro and access nodes; where the top level represents the *core nodes*. Core nodes are distributed across what so-called $Central\ Points-of-Presence$ (POPs), which are usually located in the big cities. Each Central POP hosts a pair of core nodes. Core nodes in each one Central POP are connected to each other, and to other core nodes in the adjacent cities usually by two links for failure protection. A high-capacity *Internet peering router* is connected to the core nodes to offer connectivity to the Internet. Having said that, to get connected to the Internet, there might be a number of traversed core nodes until Internet connection is established via a Central POP.

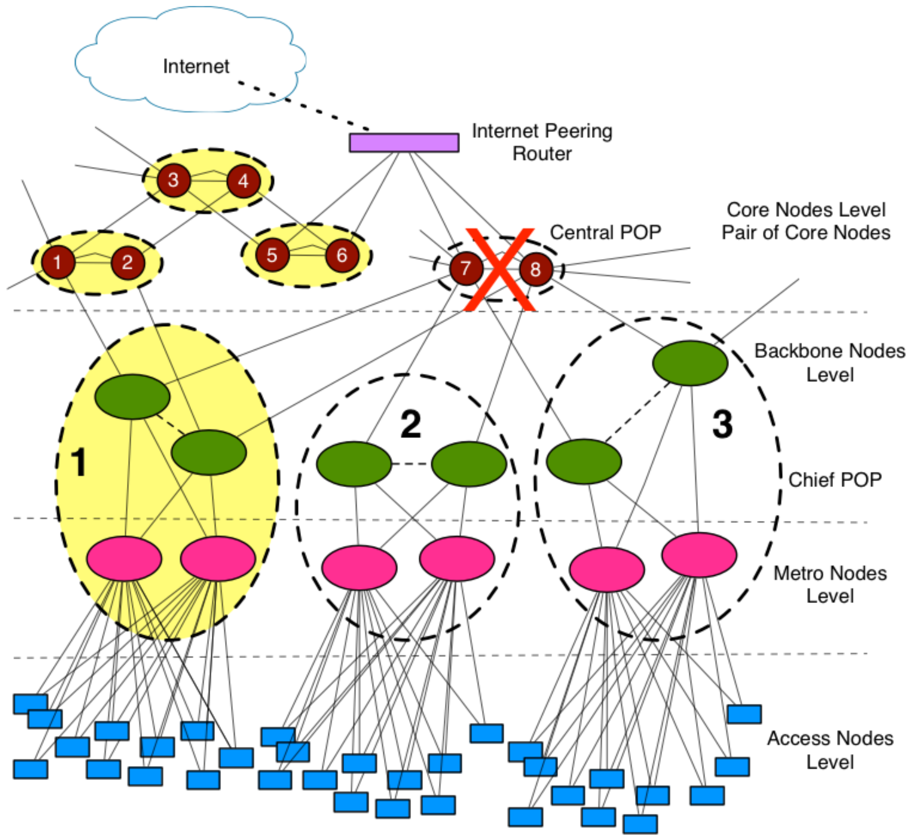

Figure 4: Hierarchal Topology of an Italian ISP

The backbone nodes are located in the second level of the topology. Each backbone node is connected to two Central POPs. Backbone nodes are located in larger POPs, which are called *Chief POPs*, spread in large city. From the other side, the backbone nodes are connected to *metro nodes*. In the same vein, each metro node is dual-homed to two backbone nodes for failure protection. Metro and backbone nodes are located in the same Chief POP. At the bottom of the hierarchal topology, we can see the access nodes, which provide connectivity to the Digital Subscriber Line Access Multiplexers (DSLAMs) to which users are connected to via DSL, FTTN, or PON as per Figure 2 in this paper. Access nodes aggregate traffic from users in the same area/vicinity. Each access node is dual-homed to the closest pair of metro nodes. As such, by using the above topology, Figure 4, failure in any of the intermediate nodes will result in redirecting the user's job into one of the live connected nodes to the failed one, based on which level in the topology the failure occurred. That means, the power consumption of each route can be vary. Hence, we need to know the

Table 1: Route A Network Components

Type	Equipment	Capacity	Power Consumption
Ethernet Switch (small)	Cisco 4507R-E	64 Gbps	0.658 kW
Ethernet Switch	Cisco 6509-E	180 Gbps	2.279 kW
BNG	Juniper E320	320 Gbps	3.347 kW
Provider Edge	Cisco 12816	160 Gbps	4.21 kW
Core Router	Juniper T640	640 Gbps	6.283 kW
WDM (800km)	Fujitsu 7700	40 Gbps	136 W/channel

energy consumption and the actual capacity of each equipment used in all possible routes starting from the access nodes level all the way to the Internet, then to the data centre in a cloud scenario, as it was discussed in *section* 4.1. As such, we consider to use the specification (power and capacity) of real network equipment provided by the manufacturers as shown in Table 1, Table 2, Table 3, and Table 4 and apply them on the above topology.

6.2. Energy evaluation and results

In this scenario, there are three standard routes leading to one of the green cloud data centres. These three routes have different structure, in terms of number of nodes, power and capacity, depending upon the Chief POP and Central POP traversed, as follows:

1. Route A (Table 1) composed of 8 core routers/nodes, 52 edge routers, 52 access routers, 260 residential switches, and 260 end hosts; hence a total of 632 nodes.
2. Route B (Table 2) composed of a smaller number of intermediate nodes compared to route A. It includes: 6 core routers, 48 edge routers, 47 access routers, 245 residential switches, and 260 end hosts, hence a total of 606 nodes.
3. Route C (Table 3) composed of the least number of intermediate nodes including: 5 core routers, 45 edge routers, 45 access routers, 230 residential switches, and 260 end-hosts, hence a total of 585 nodes.

However, as mentioned earlier, nodes failure can happen anytime during the sending and receiving process, which results in selecting, switching to, different routes to complete the process. For example, a Central

Table 2: Route B Network Components

Type	Equipment	Capacity	Power Consumption
Ethernet Switch (small)	Cisco 4503	64 Gbps	0.474 kW
Ethernet Switch	Cisco 6509	160 Gbps	3.8 kW
BNG	Juniper E120	120 Gbps	1.638 kW
Provider Edge	Cisco 12816	160 Gbps	4.21 kW
Core Router	Cisco CRS-1	640 Gbps	10.9 kW
WDM (800km)	Fujitsu 7700	40 Gbps	136 W/channel

Table 3: Route C Network Components

Type	Equipment	Capacity	Power Consumption
Ethernet Switch (small)	Cisco 4503	64 Gbps	0.474 kW
Ethernet Switch	Cisco 6509	160 Gbps	3.8 kW
BNG	Cisco ASR 9001-S	60 Gbps	3.3 kW
Provider Edge	Cisco 12816	160 Gbps	4.21 kW
Core Router	Cisco CRS-1	640 Gbps	10.9 kW
WDM (800km)	Fujitsu 7700	40 Gbps	136 W/channel

POP node of Route A, which is the red-crossed node in Figure 4, drops out of the network due to hardware failure. As a result, the backbone nodes will switch to a different Central POP, which is the yellow path in the same figure that belongs to Route B, to get connected to the Internet. By looking at the new yellow route, namely Route D in this example, it is clear that the number of the Central POP is 3-times more than the Central POP required by the original route. That logically means, applying ET_{send} on the new route results different energy consumption that might be less or more the the original one. As such, Route D (Table 4) structure comprised of 8 core routers (all from route A), 52 edge routers (all from route A), 251 residential switches (150 from route A and 101 from route B), 47 access routers (all from route B), and 260 end hosts

Table 4: Route D Network Components

Type	Equipment	Capacity	Power Consumption
Ethernet Switch (small)	Cisco 4503	64 Gbps	0.474 kW
Ethernet Switch (Route A)	Cisco 6509-E	180 Gbps	2.279 kW
Ethernet Switch (Route B)	Cisco 6509	160 Gbps	3.8 kW
BNG	Juniper E120	120 Gbps	1.638 kW
Provider Edge	Cisco 12816	160 Gbps	4.21 kW
Core Router	Juniper T640	640 Gbps	6.283 kW
WDM (800km)	Fujitsu 7700	40 Gbps	136 W/channel

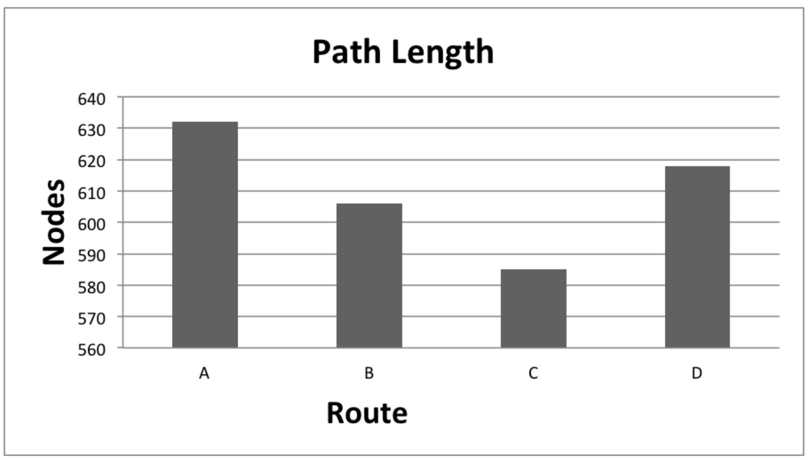

Figure 5: Path lengths of each route

(all from route B); hence the total is 618 nodes.

Figure 5 depicts the total number of the intermediate nodes to the data centre for each of the four routes in this scenario. On calculating the total energy transportation cost for the above four network settings, it is clear, as depicted in Figures 6 and 7, that although Route A has more traversed nodes than Routes B, C and D, it consumes less energy than either of these routes. This is due to the relative capacity and the power consumption of the equipment used in each of the routes. It is vital to mention that the values provided in Figure 6 are based on ET_{send} calculations of the routes, and it is assumed to be nearly the same for ET_{recv},

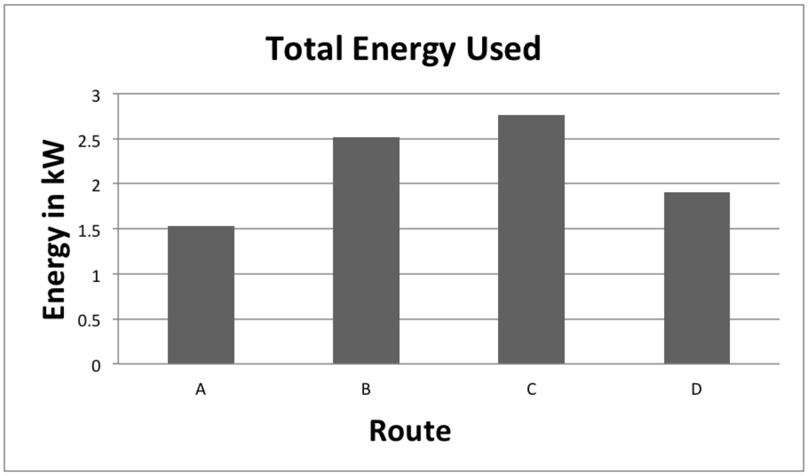

Figure 6: Total Energy of each route

if the same route has been used for inverse sending. Hence it can be seen that Route A is the most energy efficient and thus the best routes out of the four from a pure energy consumption perspective, while Route B represents a potential compromise combination of shortest path and energy consumption, and Route D can be looked at as a recovery route. Further energy efficient related results based on average energy consumption per node of each route, as outlined in Figure 7, route A despite being the longest path, is not only the least energy demanding route overall, but in addition is the most energy efficient route in that it has the lowest average energy consumption per node. This is not seen in route B, which although is less energy demanding than route C , overall, and is a shorter path than route A, is not efficient in term of energy consumption per node; the energy advantages of route B are outweighed by energy cost of its additional nodes. Therefore it can be argued that Route A is the most favourable route from an energy efficiency point of view.

7. Conclusion

Energy efficiency has become a high priority aim in cloud network environment, including data centres sustainability. Since the data centres consume the largest amount of energy in the entire cloud network, it has been greatly dealt with and solutions were already implemented and approved to get the green data centre. This paper dealt with the energy efficiency of cloud routing rather than data centres energy consumption, and proposes and evaluates a new energy efficient routing framework, dubbed GreeDi. A formal analysis of the cloud network connectivity has been given via situation calculus. GreeDi algorithm was evaluated on a

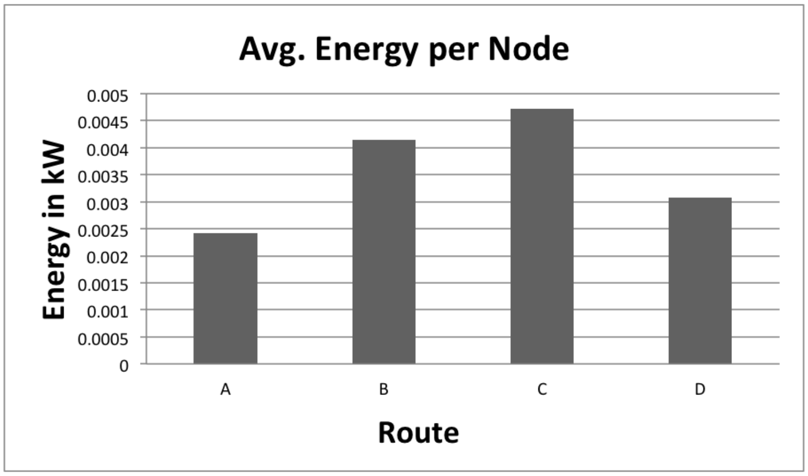

Figure 7: Average Energy consumption per node of each route

physical Italian ISP topology, that has three different routes to a green cloud data centre. From the example results shown in this paper, the shortest path approach is different from the energy efficient one, and thus, the energy efficient path is used to conform to the environmental objectives. As in any other network, nodes failure can happen at anytime during the sending and receiving process; which can lead to different energy and power consumption. As such, the number of intermediate nodes, capacity and power consumption of each node between the user and the cloud data centre have a direct impact on ET_{send} and ET_{recv}, as shown in the experiment results. The results gained from the evaluation section in this paper imply that a decision on which route is most energy efficient one can only be made after each successful sending and/or receiving process to make sure that the calculation has been done based on the traversed nodes.

Future extensions to this work include analysing and taking into account the time required for transportation, and energy and time required for computation, between the data centres and users and among the data centres themselves, in order to establish and evaluated how the proposed algorithm performs in terms of computation consumption.

References

[1] E. Gleeson. Computing industry set for a shocking change [online] (07 2013).

[2] T. A. Luxembourg. Vat aspects of cloud computing in luxembourg [online] (07 2013).

[3] K. Flinders, University of salford makes £5.7m cloud computing investment (07 2013).
URL http://www.computerweekly.com/news/1280095022/University-of-Salford-makes-57m-cloud-computi

[4] L. Zyga. How energy-efficient is cloud computing? [online] (07 2013).

[5] L. C. Yee, Google to invest 300*millioninhongkongdatacenter* (07 2013).
URL http://www.reuters.com/article/2011/12/08/us-google-hongkong-idUSTRE7B70BU20111208

[6] P. Seybold. Cloud computing/mobile computing/network bandwidth, green data centers and telehousing [online] (07 2013).

[7] J. G. Olivier, G. Janssens-Maenhout, J. A. Peters, J. Wilson, Long-term trend in global co2 long-term trend in global co2 emissions, Background study, PBL Netherlands Environmental Assessment Agency and JRC European Commission (2011).

[8] P. Baer, Exploring the 2020 global emissions mitigation gap (2008).
URL http://www.ippr.org/uploadedFiles/globalclimatenetwork/Exploring_the_Mitigation_Gap%5b1%5d.pdf

[9] S. Kuma, R. Buyya, Green Cloud Computing and Environmental Sustainability, John Wiley and Sons, Ltd, 2012, pp. 315–339. doi:10.1002/9781118305393.ch16.
URL http://dx.doi.org/10.1002/9781118305393.ch16

[10] X. W. Jinhua Zhu, Chunming Qiao, A comprehensive minimum energy routing scheme for wireless ad hoc networks, in: The 23rd Conference of the IEEE Communications Society: IEEE INFOCOM 2004, IEEE, 2004.

[11] X. W. Jinhua Zhu, Model and protocol for energy-efficient routing over mobile ad hoc networks, IEEE Transaction on Mobile Computing 10 (11) (2011) 1546–1557.

[12] A. M. Suman Banerjee, Minimum energy paths for reliable communication in multi-hop wireless networks, in: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking computing: MobiHoc'12, ACM New York, NY, USA, 2002, pp. 146–156.

[13] N. B. Keith Scott, Routing and channel assignment for low power transmission in pcs, in: 5th IEEE International Conference on Universal Personal Communications, Vol. 2, IEEE Xplore, 1996.

[14] T. H. M. Volkan Rodoplu, Minimum energy mobile wireless networks, IEEE JOURNAL ON SELECTED AREAS in Communications 17 (8) (1999) 1333–1344.

[15] Z. L. C. G. M. C. L. Zhang, C. Wu, F. C. M. Lau, Moving big data to the cloud, in: Proceedings IEEE INFOCOM, 2013, pp. 405–409.

[16] Z. L. C. G. M. C. L. Zhang, C. Wu, F. C. M. Lau, Moving big data to the cloud: An online cost-minimizing approach, in: IEEE J. Sel. Areas Commun, Vol. 31, 2013, pp. 2710–2721.

[17] R. F. M. Gattulli, M. Tornatore, A. Pattavina, Low-carbon routing algorithms for cloud computing services in ip-over-wdm networks, in: IEEE International Conference on Communication (ICC), 2012, pp. 2999–3003.

[18] H. Kantarci, B.; Mouftah, Optimal reconfiguration of the cloud network for maximum energy savings, in: 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid),, 2012, pp. 835–840.

[19] L. Liu, H. Wang, X. Liu, X. Jin, W. B. He, Q. B. Wang, Y. Chen, Greencloud: a new architecture for green data center, in: Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session, ICAC-INDST '09, ACM, New York, NY, USA, 2009, pp. 29–38. `doi:10.1145/1555312.1555319`.
URL `http://doi.acm.org/10.1145/1555312.1555319`

[20] J. Conejero, O. F. Rana, P. Burnap, J. Morgan, C. Carrion, B. Caminero, Characterising the power-performance trade off in hadoop clouds: A social media analysis case study, in: 3rd International Conference on Cloud Computing and Services Science, Vol. To appear, 2013, pp. 1–12.

[21] H. N. Van, F. D. Tran, J.-M. Menaud, Performance and power management for cloud infrastructures, in: IEEE International Conference on Cloud Computing, CLOUD 2010, Miami, FL, USA, 5-10 July, 2010, IEEE, 2010, pp. 329–336.

[22] G. Laszewski, L. Wang, Greenit service level agreements, in: P. Wieder, R. Yahyapour, W. Ziegler (Eds.), Grids and Service-Oriented Architectures for Service Level Agreements, Springer US, 2010, pp. 77–88. `doi:10.1007/978-1-4419-7320-7_8`.
URL `http://dx.doi.org/10.1007/978-1-4419-7320-7_8`

[23] H. T. H. K. T. Satoh, F.; Yanagisawa, Total energy management system for cloud computing, in: IEEE International Conference on Cloud Engineering (IC2E), 2013, pp. 233–240.

[24] C.-C. Lin, P. Liu, J.-J. Wu, Energy-efficient virtual machine provision algorithms for cloud systems, in: IEEE 4th International Conference on Utility and Cloud Computing, UCC 2011, Melbourne, Australia, December 5-8, 2011, IEEE Computer Society, 2011, pp. 81–88.

[25] H. Al-Daoud, I. Al-Azzoni, D. G. Down, Power-aware linear programming based scheduling for heterogeneous computer clusters, Future Generation Comp. Syst. 28 (5) (2012) 745–754.

[26] T. Baker, O. F. Rana, R. Calinescu, R. Tolosana-Calasanz, J. A. B. nares, Towards autonomic cloud services engineering via intention workflow model, in: 10th International Conference on Economic of Grids, Clouds, Systems and Services, Vol. 8193, 2013, pp. 212–227.

[27] B. N. R. E. B. M. A. P. M. T. A. K. M. Kaoru Kawamoto, Jonathan G. Koomey, Electricity used by office equipment and network equipment in the us: Detailed report and appendices, in: ACEEE Summer Study Conf. Energy Efficiency in Buildings, LBNL-45917, 2000.

[28] S. L. C. Barthel, S. Thomas, Ghg emission trends of the internet in germany, in: International Climate Policy and IT Sector Joint Workshop with IGES and the Wuppertal Inst., 2001.

[29] T. Volker, Assessing the resource intensity of the internet infrastructure: Data analysis for a material-flow oriented approach and first results on electricity consumption., IIIEE, M. Sc. thesis at the Lund University, Sweden.

[30] B. N. S. Lanzisera, R. Brown, Data network equipment energy use and savings potential in buildings, ACEEE Summer Study on Energy Efficiency in Buildings (2010).

[31] H. K. S. W. T. R. Baliga J., Ayre R.W.A., Energy consumption in optical ip networks, Journal of Lightwave Technology 27 (13) (2009) 2391–2403.

[32] K. S. G. S. V. P. S. D. B. O. Kilper D.C., Atkinson G., Power trends in communication networks, IEEE Journal of Selected Topics in Quantum Electronics 17 (2) (2010) 275–284.

[33] M. F. R. A. R. S. T. Kerry Hinton, Jayant Baliga, Power consumption and energy efficiency in the internet, IEEE Network 25 (2) (2011) 6–12.

[34] J. Baliga, R. Ayre, K. Hinton, R. Tucker, Green cloud computing: Balancing energy in processing, storage, and transport, Proceedings of the IEEE 99 (1) (2011) 149–167.

[35] H. Levesque, F. Pirri, R. Reiter, Foundations for the situation calculus, Linkoping Electronic Articles in Computer and Information Science 3 (18).

[36] R. Cohen, S. Havlin, D. ben Avraham, Efficient immunization strategies for computer networks and populations, Physical Review Letters 91 (24) (2003) 5.

[37] R. T. C. O. R. M. R. Thar Baker, Yanik Ngoko, Energy efficient cloud computing environment via autonomic meta-director framework, in: Sixth International Conference on Developments in eSystems Engineering (DeSE'13), IEEE, 2013, pp. 198–203.

[38] Spring-nextel applied research group ipmon project (January 2005) [cited 10 Dec 2014].
URL http://ipmon.sprint.com

[39] L. Ceuppens, Planning for energy efficiencyvnetworking in numbers, in: Optical Fiber Communication Workshop Energy Networks., 2009.

[40] C. Panarello, A. Lombardo, G. Schembra, L. Chiaraviglio, M. Mellia, Energy saving and network performance: a trade-off approach, in: e-Energy'10 Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, ACM New York, NY, USA, 2010, pp. 41–50.