Simulating Power Saver Network Links in MATLAB

Huseyin Abaci
Adnan Menderes University,
Computer Engineering, Aydın/Turkey,
huseyin.abaci@adu.edu.tr

Gerard Parr, Sally McClean, Adrian Moore
University of Ulster,
School of Computing & Engineering,
Coleraine, NI, UK, BT52 1SA
{gp.parr, si.mcclean,
aa.moore}@ulster.ac.uk

Louise Krug
BT Group, Adastral Park
Martlesham, Ipswich, IP5 3RE
louise.krug@bt.com

ABSTRACT
Network devices, meeting increasing workload demand, are not efficiently power-workload proportionate and consume a considerable amount of power even when the workload (utilisation) is low. Proposed Slowing Mechanism (SM) provides power workload proportionality for a wired network equipment to reduce power consumption. In order to implement the SM we wrote a simulation in MATLAB to analyse the feasibility of the solution. The simulation provides an insight into how a power saving technique can be employed in a network environment and how the parameters of hardware and ICT applications are interlinked with each other. The simulation creates incoming traffic, packet processing, buffer occupancy, packet delay, throughput, and outputs power consumption and power overhead of the SM.

Categories and Subject Descriptors
C.2.1 [**Network Architecture and Design**]: Circuit-switching networks, Network communications, Network topology, Packet-switching networks; C.2.3 [**Network Operations**]: Network management, Network monitoring; C.2.6 [**Internetworking**]: Routers; C.4 [**Performance of Systems**]: Design studies, Measurement techniques, Modeling techniques, Performance attributes, Reliability, availability, and serviceability; I.6.4 [**Model Validation and Analysis**]; I.6.5 [**Model Development**]: Modeling methodologies.

General Terms
Algorithms, Management, Measurement, Performance, Design, Experimentation.

Keywords
Simulation, Communications Network, Power Saving, Energy, QoS.

1. INTRODUCTION
The associated improvement allows and encourages the developers deploying the ICT services to offer better facilities and services. Examples of the types of services provided include bandwidth intensive Video-on-Demand (VoD), Voice over Internet Protocol (VoIP), cloud computing, online multi-player gaming and file sharing. Previous estimates show that as a consequence of increasing Internet scale, ICT consumes 2% - 3% of the total global energy consumption [1]. Experts estimate that Internet Protocol (IP) traffic will continue to grow at 43% per annum, doubling every 1.4 years [2][3]. This pervasive computing, especially, has created increasing pressure on the network infrastructure due to the requirement for high volume data transmission between remote ends. Network devices, meeting increasing workload demand, are not efficiently power-workload proportionate and consume a considerable amount of power even when the workload (utilisation) is low [4][5][6]. The results in [5] show that there is no wide application of power workload proportionality in general switching and routing network equipment. In consideration of all the above, urgent power-workload optimisation of the network infrastructure is required in order to establish more energy efficient networks by employing state-of-the-art power saving technologies [7].

This work proposes a Slowing Mechanism (SM) that provides power workload proportionality for a wired communication link of the network equipment to reduce power consumption [9]. The slowing will be achieved by a Dynamic Voltage-Frequency Scaling (DVFS) technique that increases or decreases votage-frequency to adjust the Operational Rate (OPR) that is the forwarding packets/bits rate of a link adjusted according to the workload. However, performance and power are trade-offs, and the slowing power saving technique comes at the cost of network performance degradation. To meet applications' (VoIP, Data, and Video) performance requirements, a Safety Gap (SG) is proposed in the SM.

In order to implement the SM we wrote a simulation to analyse the feasibility of the solution. Many parameters need to be carefully set for performance requirements within SM. The optimisation of the required hardware and SM parameters e.g. SG, packet processing cycle, max frequency are determined by a Genetic Algorithm (GA) optimisation tool dynamically set to respond to the variable incoming traffic pattern. The simulation is written in MATLAB v7.9.0 with over 2000 code lines. A Genetic Algorithm and Direct Search Toolbox (codes are also modified by authors) was used within the simulation for optimising the parameters. Simulation allows for the simulation of hardware characteristics of the routers and acts as if it is a Device Under Test (DUT) that is capable of providing the SM and reporting of performance degradation and power consumption. This provides an insight into how a power saving technique can be employed in a network environment and how the parameters of hardware and ICT applications are interlinked with each other. To the best of our knowledge there is no router, which performs the SM. In addition, measuring the power consumption and performance degradation of a DUT is yet another challenge. With the simulation, we could change and modify the parameters that neither the OPNET modeller nor real routers are able to perform. Therefore, the simulation provides

SIMUTOOLS 2015, August 24-26, Athens, Greece

DOI 10.4108/eai.24-8-2015.2261118

flexibility for us and creates Incoming Traffic (InT), packet processing, buffer occupancy, packet delay, throughput, and outputs power consumption and power overhead of the SM. Generally, the simulation covers the SM functions.

The structure of this paper is as follows: Section 2 presents Quality of Service (QoS) and algorithm of SM, Section 3 explains the simulation methodology and SM's equations that are used in the Simulation, Section 4 contains related work and Section 5 contains conclusion and future direction of this work.

2. QOS AND ALGORITHM OF SM

Three performance metrics has been considered within the SM, throughput, packet delay, packet loss. It is relatively easy to meet throughput and packet loss QoS requirements if the SM closely observes the buffer occupancy and carefully selected buffer thresholds. On the other hand, delay is a highly variable, important performance metric and it is difficult to calculate and measure the E2E delay of a real network environment. The degradation of the delay explicitly affects performance and Quality of Experience negatively. Moreover, sudden bursts are difficult to predict and may cause performance degradation. According to the size and duration of the burst, buffer occupancy increase, adds delays, and packet drops may occur due to excessive buffering. Therefore, the SM should observe buffer occupancy and set the frequency level higher when the buffer threshold is exceeded to avoid packet drop. However, although the system does not degrade throughput nor cause packet drop, there is still a certain amount of delay inevitably introduced by the SM. A general principle of the SM is in the reduction of OPR (service rate) to get closer to InT (arrival rate); usually this results in a utilisation of between 60% and 90%. Utilising device components at higher levels causes extra buffering and increases the time to process InT amount of data and consequently packet delay [15][16][30]. Therefore, the delay metric was investigated rather deeply, because this metric has greater importance in the SM.

2.1 Algorithm of SM

An algorithm is assumed within the SM that has three logical components: InT, SG and OPR; our energy-saving scheme is developed based on them, as shown in Figure 1. The InT is the number of packets (load) that enters the link per Sampling Interval (SI) and InT flow varies by time. The SI is a time interval (a bin) in which each bit or packet is counted. Thus, the number of packets or bits arrives at the link in that interval is called a "sample" in this paper.

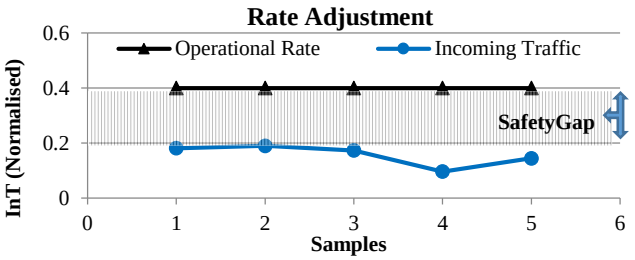

Figure 1. Three logical components of the SM, Incoming Traffic (Load), Safety Gap and Operational Rate (OPR). Maximum capacity of link is 1 (normalised) and not shown in the plot.

As we said earlier to determine (predict) the required OPR without affecting QoS, the SM needs to observe the InT closely. For this reason, in the SM, one of simplest and most popular forecast methods, EWMA, is used to perform functions such as monitoring, controlling and predicting future InT. The EWMA acts as a sub-mechanism of the SM that increases and decreases the OPR without requiring complex feedback monitoring by keeping required SG between the InT and the OPR. EWMA is a standard [13] for all-time series based forecasting due to its simplicity, accuracy, computational efficiency and flexibility.

2.2 Integration of Slowing Mechanism's Components

If maximum forwarding capacity of a link is C^{max} in bits per second "bps" or in packets per second "pps" and $OPR = C^{max}$, then the is no slowing in operation. As mentioned earlier, the SG is the amount of safety gap needed in order to compensate for fluctuation of bitrates, compensate transition delay and provide smaller delay. The following equation calculates the SG:

$$SG = sg * C^{max} \quad (1)$$

where SG is in "bps" if C^{max} is in "bps" or SG is in "pps" if C^{max} is in "pps" and sg is a safety gap fraction of total capacity which is $0 \leq sg < 1$. Every SI time, the SM monitors the data flow to detect any sudden increase in the rate. A single sample may not provide enough information for traffic pattern analysis. Therefore, OI is the number of samples (SIs) required to be collected for statistics and future prediction. OI may comprise of one or more SIs and $SI \leq OI$. The statistics are collected regularly by the SM itself and the decision mechanism is based on the data gathered during OI. This means that the SM does not need to retain any traffic characteristics longer than the OI. However, setting this parameter requires attention as this has an important impact on the performance and power saving [10][8]. The OPR is the SM's forwarding rate and fraction of C^{max} that is adjusted by frequency. The OPR is $OPR < C^{max}$ if slowing is in operation, otherwise $OPR = C^{max}$. The OPR is calculated using (1) and the following equation.

$$OPR = Z_i + SG \quad (2)$$

the Z_i is the EWMA at i^{th} sample. Let us assume that w is a weight and $0 \leq w \leq 1$, μ_0 is the historical mean that is the average of the samples within the first OI, samples of size $n \geq 1$ are collected and $\bar{x}_i$ is the average of a sample at i. Assume that $z_0 = \mu_0$ and the exponentially weighed moving average (Z_i) [14] is:

$$Z_i = w\bar{x}_i + (1 - w)Z_{i-1} \quad (3)$$

Each older sample's weight is decreased by a factor of $(1 - w)$. The value of w determines how sensitive the response is to the mean value. If the weighting is high, the last sample is the most important, the system will be sensitive to any sudden change, and the EWMA will almost track the sample measurements. Therefore, we gathered the following equation using (1), (2) and (3) in order to calculate OPR for the next sample.

$$OPR = w\bar{x}_i + (1 - w)Z_{i-1} + sg * C^{max} \quad (4)$$

Determining w and sg requires extra attention. These values vary according to traffic pattern, and the same values may result in different saving and performance. The SM requires an optimisation algorithm for these variables. The GA has been used for this purpose and details of GA optimisation is not in scoop of this paper.

Overall, the SM observes incoming traffic, sets OPR (higher or lower) by providing sufficient safety gap according to incoming traffic, observes performance metrics and if performance degradation occurs, takes appropriate action. A flow chart and functional blocks of the simulation are shown in Figures 2 and 3. In addition, operation of the simulation is shown in Figure 4 that is based on time vector.

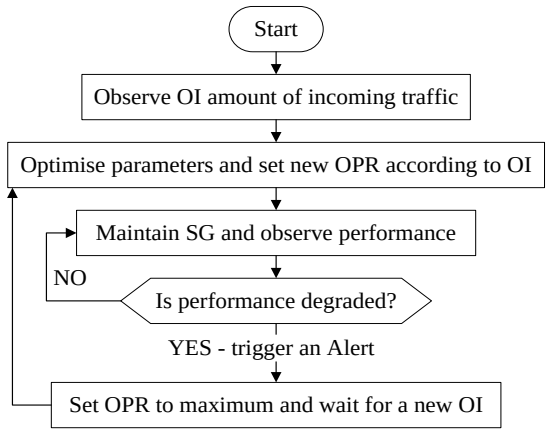

Figure 2. Flow chart shows working principle and algorithm of the SM. The SM runs GA first to optimise parameters and checks the performance regularly.

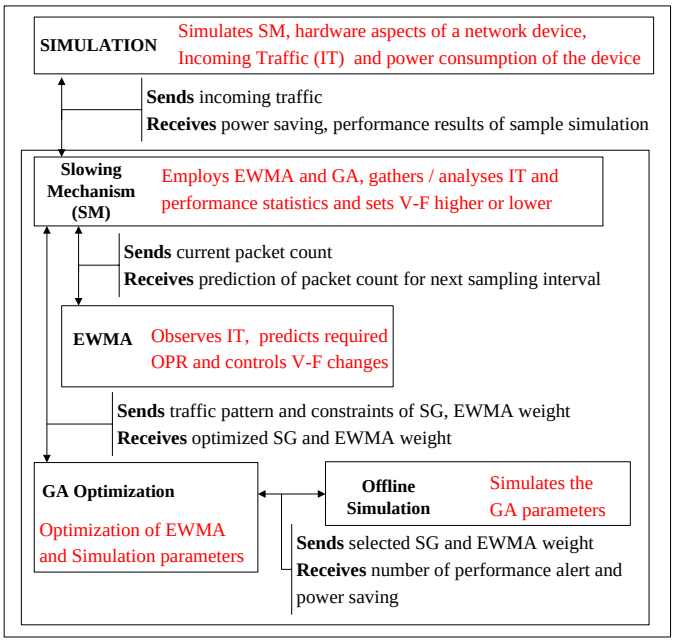

Figure 3. Functional blocks of the simulation.

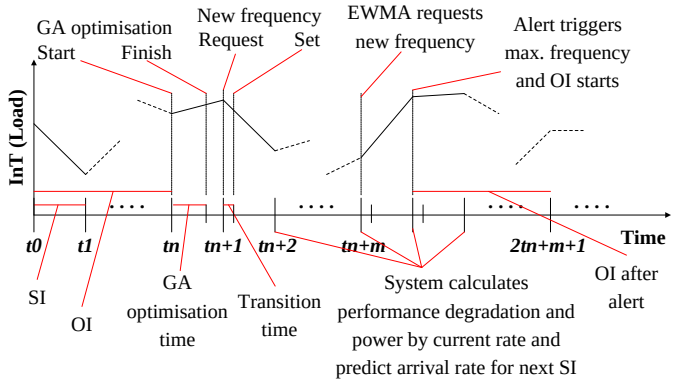

Figure 4. The SM operation with respect to time and load. The figure shows the SI, OI, optimisation time, transition time, and illustrates GA optimisation, frequency changes according to InT and performance.

The power saving and the performance analysis belongs to the target port card's transmission controller and this is shown in Figure 5. Here is a Router A connected to Router B via physical link, and according to load the SM reduces the OPR. This may introduce acceptable End-to-End (E2E) delay.

2.3 Application and Per Hop Delay Analysis

In this section, the maximum allowable delay per packet of a delay constrained application is found. This is called a delay threshold in the paper and denoted by the symbol D^{thrs}.

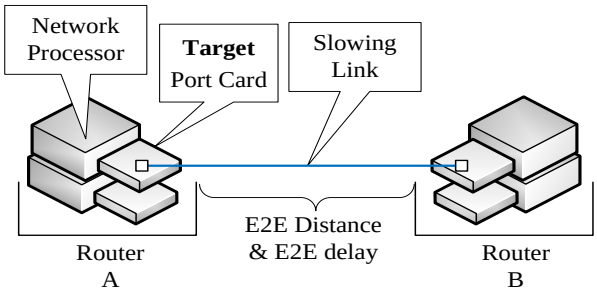

Figure 5. Simplified architecture of slowing link. Router to router connection shows E2E (End-to-End) delay, distance, and power saving belongs to target port cart.

Based on references [17][18][19] that the tightest delay constrained application is VoIP which requires a network delay between 75 ms and 120 ms to provide best MOS = 5 (Mean Opinion Score). Although the priorities of delay tolerant (i.e. file transfer, email) and non-delay tolerant (i.e. VoIP) applications are different at the sender buffer, they may share the same physical link and any slowing on a link may introduce noticeable delay for VoIP but this may not be noticeable for the data traffic. For this reason, our first order solution is a per hop basis decision mechanism in which each hop applies the same delay threshold. If D^{thrs} satisfies the VoIP (tightest delay tolerance), this delay threshold will therefore also satisfy other packet types - each hop should not delay a packet more than D^{thrs}. Assuming that maximum hop number between an S and D pair, N^{hops}, is around 10 [20], it is shown above that the one way acceptable network delay for the delay constraints application VoIP is 75ms. An approximate per hop delay is calculated using the following equation:

$$D^{thrs} = \frac{VoIP\ e2e\ delay}{N^{hops}}$$

where D^{thrs} is the allowed per hop delay in "ms", N^{hops} is the number of hops between S-D pair and it set to 10, $VoIP\ e2e\ delay$ is 75ms. Therefore, the allowed per hop delay is 7.5 ms ($D^{thrs} = 7.5\ ms$) and this is suitable for the best quality VoIP communication.

3. THE SIMULATION METHODOLOGY AND SLOWING MECHANISM'S EQUATIONS

3.1 OPR and Frequency Relationship

The number of packets processed per second is dictated by the operational frequency, and number of cycles required for processing a single packet. However, within the simulation there is more than one frequency and more than one OPR level is required to be calculated. The maximum OPR that the link can perform depends on the maximum frequency and is calculated by the following equation:

$$OPR^{max} = F^{max} * \frac{1000000}{T^{pckt}} \quad (6)$$

where OPR^{max} is the maximum operational rate in "pps", F^{max} is the maximum frequency of the link in MHz, T^{pckt} is number of cycles required to process a packet. The number 1000000 is used to convert cycles to MHz because T^{pckt} is in cycles. Here we used a packet cycle of 5000 [8], $T^{pckt} = 5000\ cycles$, for a packet length of 1500bytes, $L^{pckt} = 1500\ bytes$. We took the maximum packet size that travels on the link (1500 bytes) in order to cover the worst case scenario and simplify the solution. In a similar manner - the current OPR, which the link is operating on, is calculated using the following equation.

$$OPR^{now} = F^{now} * \frac{1000000}{T^{pckt}} \quad (7)$$

where OPR^{now} is current packet proccessing capability of the link in "pps" that is dictated by current frequency F^{now}in MHz. The number 1000000 is used to convert cycles to MHz because T^{pckt} is in cycles.

Moreover, OPR^{req} is requested OPR in pps by the SM according to Z_i and SG, and calculated by (4). The SM needs to find the required frequency to forward requested (predicted) number of packets to transmit. This is important to satisfy the throughput, delay, packet loss and buffering performance metrics. The EWMA predicts and the SM calculates the required OPR^{req} according to IT_i , Z_i and SG. Then the SM calculates the equivalent frequency and sets. The OPR^{req} is satisfied after setting the new frequency. The SM uses the following equation to calculate the required frequency:

$$F^{req} = \frac{OPR^{req} * T^{pckt}}{1000000} \quad (8)$$

where F^{req} is the requested frequency in MHz that transmits OPR^{req} amount of packets. Again, the number 1000000 is used to convert cycles to MHz because T^{pckt} is in cycles.

The SM may not be able to set exactly the same F^{req} requested by the EWMA, due to the SM sets the new frequency from the predetermined tabulated voltage-frequency pairs. A voltage level range from 0.7 to 1.25V [21][22] and 125MHz maximum frequency [12] are used for conservative approach, and the SM has 10 uniform voltage-frequency pairs to switch between. Table 1 shows these adjacent voltage-frequency pairs used alongside normalized frequency in the simulation. If the tabulated Voltage-Frequency (V-F) pairs are $(v,f)_j$ and number of pairs is 10 then $j = 1,2,\dots,10$, and minimum voltage frequency pair is $(v,f)_1 = (v,f)_{min} = (0.7\,V, 38.75\,MHz)_1$ and maximum voltage-frequency pair is $(v,f)_{10} = (v,f)_{max} = (1.25\,V, 125\,MHz)_{10}$. Therefore, the SM has to set one of the $(v,f)_j$ available within the V-F table and the SM cannot set a F^{now} in between the adjacent V-F pairs.

Table 1. Tabulated Voltage-Frequency (V-F) pairs used within the SM. The SM sets equal or nearest upper frequency and voltage pair from the table according to requested OPR.

Voltage-Frequency Pairs			
$(v,f)_1$	(0.70 V, 38.75 MHz)	$(v,f)_6$	(1.01 V, 86.25 MHz)
$(v,f)_2$	(0.76 V, 48.75 MHz)	$(v,f)_7$	(1.07 V, 96.25 MHz)
$(v,f)_3$	(0.82 V, 57.5 MHz)	$(v,f)_8$	(1.13 V, 106.25 MHz)
$(v,f)_4$	(0.88 V, 67.5 MHz)	$(v,f)_9$	(1.19 V, 115 MHz)
$(v,f)_5$	(0.94 V, 77.5 MHz)	$(v,f)_{10}$	(1.25 V, 125 MHz)

The SM always sets the higher pair of adjacent pairs for the sake of the performance. Therefore, the frequency and voltage pair that the link operates on is set as follows:

$$F^{now} \in (v,f)_j$$

$$\text{Subject to } (v,f)_{j-1} < F^{req} \leq F^{now}$$

$$\text{Therefore, } (v,f)_j = F^{now} = F^{req} + FG \quad (9)$$

$$and\ FG = \begin{cases} F^{now} - F^{req}, & if\ F^{req} < (v,f)_j \\ 0, & if\ F^{req} = (v,f)_j \end{cases}$$

where F^{now} is frequency of $(v,f)_j$ pair set by the SM in MHz and FG is Frequency Gap between F^{now} (OPR^{now}) and F^{req} (OPR^{req}). If $OPR^{now} = OPR^{req}$ then $FG = 0$. Figure 6 shows relationship between frequency and packet forwarding domains.

The SM should calculate the number of packets it needs to be forwarded by the link in the next time window. This is predicted by EWMA according to InT. This forwarding capacity should be translated into the frequency domain that is a low-level hardware translation of the packet per second using (6), (7), (8) and (9).

3.2 Packet Delay

In order to analyse the SM's packet delay a testbed was setup in our laboratory to perform empirical observations. The sake of brevity the details of testbed has not been explained in this paper. The results provided a realistic figure of the slowing's delay trend, and performance degradation at the destination node. Testbed results showed that when ρ (normalised utilisation, $\frac{IT}{OPR}$) approaches 1, packet delay increases. In order to formulate this trend a regression fit is used to analyse the trend shown in Figure 7. However, our initial results showed that in the SM, ρ is always in the range $0.5 < \rho < 1$ due to use of the appropriate safety gap. There are only trivial packet delay differences for the range of values $0 < \rho < 0.5$ and for this reason we eliminate the points between $0 < \rho < 0.47$. Accordingly, from the new points after the elimination, we found that the cubic polynomial least square fits very well with our data with Pearson correlation coefficient R squared value $R^2 = 0.9999$. This is shown in Figure 7. The following equation is found with least squares fit through the remaining points in order to calculate per hop packet delay, D^{hop}, in "ms":

$$D^{hop} = \begin{cases} 113\,\rho^3 - 192\,\rho^2 + 110\,\rho - 20\,, & if\ \rho \geq 0.47 \\ 0.09\ ms, & otherwise \end{cases} \quad (10)$$

Moreover, propagation delay refers to the time it takes to send a bit from one end of the link to the remote end and it is not included in (10). This depends on the length of the link and its propagation speed.

The testbed measurement includes only a short distance (<5 metres) propagation delay, D^{prop}. This delay is insignificant for a short distance E2E connection. However, if two end points are connected via a long haul link, the D^{prop} to be calculated separately. If the length of the end-to-end link is specified as Ll^{e2e} in kilometers (km), and the propagation speed as $PropSpeed$ in meters per second (m/sec), first 1000 is used to convert km to meter and following 1000 is used to convert second to ms, then propagation delay is [23]:

$$D^{prop}(ms) = \frac{Ll^{e2e} * 1000\,m * 1000\,ms}{PropSpeed\ \frac{m}{sec}} \quad (11)$$

here, Ll^{e2e}set to 0.005km as a default in the SM, and the propagation speed of coaxial and fibre cable is approximately $2 * 10^8\ m/sec$ [23].

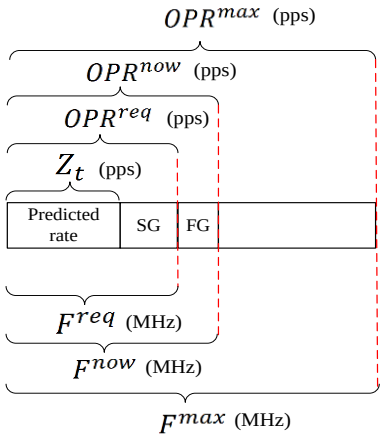

Figure 6. Illustration figure of the SM's frequency (MHz) and forwarding (pps) domain relationship. $OPR^{max} = C^{max}$ and assuming that $F^{req} < (v,f)_j$. SG is safety gap and FG is frequency gap that are added by the SM.

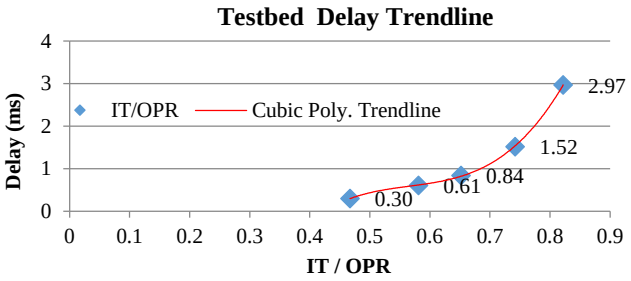

Figure 7. Cubic polynomial trend line of delay and ρ (IT / OPR). Points are results of the test-bed experiment. Cubic polynomial least square fits very well with our data points with $R^2 = 0.9999$.

Furthermore, there are two more per hop delays introduced by the SM: i) Transition time, D^{trans}, which is the time required to change between $(v,f)_{j \to j\pm1}$, and ii) Synchronisation time, D^{sync}, is the time required to link protocols and negotiate a new OPR between two end points e.g., PC-switch, switch-router at the link level this is assumed approximately equal to 1ms [24][25]. If T^{trans} is the time required for a single level change ($(v,f)_{j \to j\pm1}$) in cycles and if the level between the current frequency and requested frequency is n, the total transition time in clock cycles, $Ttot^{trans}$ between these levels, $(v,f)_{now \to requested}$ is calculated as follows.

$$Ttot^{trans} = T^{trans} * n \tag{12}$$

References [11][26] show that the DVS transition delay should be no bigger than 10 µs and this comes within the range of 1 and 100 cycles. In simulation the SM performs a decision according to the delay in "ms" at the end of the SI and transition delay in ms is represented by D^{trans}. The D^{trans} is dependent on the $Ttot^{trans}$. Consequently, T^{trans} (see (12)) and F^{now}, this was formulised using the following equation:

$$D^{trans} = \frac{Ttot^{trans}}{F^{now} * 1000000} * 1000 \tag{13}$$

where the $Ttot^{trans}$ is calculated using (12), number 1000000 is used to convert F^{now} from MHz to cycles because $Ttot^{trans}$ is in cycles and the number 1000 used to convert "second" to "ms" (F^{now} is number of cycles per sec.). This is obvious from the equation that slowing and T^{trans} has direct effect on the D^{trans}.

Finally, the total per hop delay D^{hop} in "ms" at the end of each SI is calculated using (10), (11), (12) and (13). Note that if no slowing happens within the target SI, D^{trans} and D^{sync} are "0" and excluded from the equation shown below.

$$D^{hop} = \begin{cases} D^{hop} + D^{prop}, & if\ Ttot^{trans} = 0 \\ D^{hop} + D^{prop} + D^{trans} + D^{sync}, & if\ Ttot^{trans} \neq 0 \end{cases} \tag{14}$$

3.3 Buffering

Today's highly capable metro-core routers are supported by a large amount of buffering and it is not unusual to have a packet buffering capacity of more than 1000MB [27]. The SM increases the requirements for buffering and buffer occupancy needs to be investigated closely. Slowing, limitation of prediction, rate transition and sudden bursts may cause the buffer to fill up easily at the metro-core router domain.

Therefore, the outstanding packets (extra packets) are stored in the buffer which happens if $IT_i > OPR_i^{now}$, and this buffering is denoted by B_i^{outP}. Equation 15 is used below and found B_i^{outP} due to the possible limitation of the prediction which is calculated from the difference between current forwarding capacity OPR_i^{now} and incoming traffic (arrival rate) IT_i:

$$B_i^{outP} = OPR_i^{now} - IT_i$$
$$\text{and } B_i^{outP} = \begin{cases} |B_i^{outP}|, & if\ B_i^{outP} < 0 \\ 0, & otherwise \end{cases} \tag{15}$$

Moreover, if a state transition has been made before current SI, there is a short pause in packet forwarding. This increase in buffering is calculated using the following equation.

$$B_i^{trans} = \frac{Ttot^{trans}}{T^{pckt}} \tag{16}$$

where B_i^{trans} is buffering caused by state transition in pps and $Ttot^{trans}$ is total transition cycles using (12).

Furthermore, the SM should calculate the current buffering with respect to ρ. Let as assume that buffer size is B^{size} and B^{now} is current number of packets in the buffer and current buffered packets is calculated using equation in [28] with a little modification to give the result in "packets" rather than "bits".

$$B_i^{now} = \frac{(2 * D^{hop} * \frac{1}{1000}) * OPR^{now}}{\sqrt{N^{Flows}}} \tag{17}$$

here the RTT delay of the equation in [28] is double the one way delay which makes it $2 * D^{hop}$ in the SM, and the number 1000 is used to convert D^{hop} from ms to seconds because OPR^{now} is in pps. N^{Flows} is the number of flows that share the same egress port. Although N^{Flows} is a variable, the average of N^{Flows} was taken such that $N^{Flows} = 10000$ [28] within the SM. However, this equation does not represent buffering for slowing the link on its own; B_i^{trans} and B_i^{outP} should be added to the equation. Therefore, we can calculate total buffering in Mbit per SI_i with (15), (16) and (17) as shown below.

$$B_i^{now} = \left(B_i^{now} + B_i^{trans} + B_i^{outP}\right) * 0.012 \tag{18}$$

here, the number 0.012 is single packet size in Mbit (1500 bytes = 0.012 Mbit).

Finally, real metro-core router is over provisioned, if the buffer size is B^{size}, $B^{size} > B_i^{now}$ and set to 128Mbit as a default in the simulation. If $B_i^{now} > B^{size}$ then packet loss occurs and this is calculated using the following equation.

$$PLoss^{tot} = \frac{\sum_{i=1}^{T^{sim}} PLoss_i}{\sum_{i=1}^{T^{sim}} IT_i} * 100 \quad (19)$$

where $PLoss^{tot}$ is total packet loss during simulation in percent (where 100 is used in the equation), T^{sim} is simulation time in SI, $PLoss_i$ is number of packets loss at the i^{th} sample in pps and IT_i is number of packets arrived at the i^{th} sample in pps.

3.4 Rate Switching

There are three different mechanisms used within the SM for increasing and decreasing the OPR^{now}. The SM normally increases and decreases the OPR^{now} under normal operation. However, an exceptional situation can occur with sudden spikes and high SG. Generally increasing is required when a burst is detected and this requires an urgent response as this seriously affects the D^{hop} and the B^{now} (performance). Therefore, the SM checks if these two metrics are over the threshold level or not. The D^{thrs} is set to 7.5 ms, buffering threshold, B^{thrs}, set to 80% of the B^{size}. The simulation checks if $D^{hop} > D^{thrs}$ or $B^{now} > B^{thrs}$ and if either of them is true then the SM is required to increase OPR^{now}. We set two thresholds for decreasing, i) $sgHold^{thrs}$ which checks if sg is large or not and $sgHold^{thrs} = 0.1$ as a default in the SM. If $sgHold^{thrs} \leq 0.1$, the power waste is at a minimum therefore no need for an extreme rate reduction by the GA optimisation. The EWMA prediction is able to provide the required performance and power saving, because the w and sg parameters are optimised by the GA, ii) $GAhold^{thrs}$ that is the threshold sample interval between the GA runs and $GAhold^{thrs}$ is set to a minute in the SM as a default. A minute periodic time interval is not unusual practice for the networking. More than a minute may causes extra power waste and less than a minute may causes performance degradation. This is equivalent to $GAhold^{thrs} = 60$ SI if a SI is one second. If $GAInterv$ is samples between previous GA run and current time *t*, then the simulation runs the GA if $GAhold^{thrs} \leq GAIntrv$ and $sg > sgHold^{thrs}$. These thresholds prevent frequent GA optimisation run as this leads to more resource usage, power consumption and performance degradation. Initial values showed that the default values above successfully provide power saving and the required performance.

3.5 Power Consumption and Saving

The simulation calculates two types of power, i) Power consumption of a Slowing Link, P^{SL}, which is generated by the SM with the associated power overheads and ii) power consumption of a non-Slowing Link, P^{nSL}, where the SM has not been employed. However, the overhead of the SM is low and is easily compensated for in a few seconds during OPR reduction. Due to over provisioning at the metro-core router the OPR is less than the maximum forwarding capacity, $OPR^{now} < C^{max}$.

In the simulation a well-known canonical equation is used to calculate power [29] and the power dissipated by non-Slowing Link in an i^{th} sample , P_i^{nSL}, and Slowing Link, P_i^{SL}, are shown by,

$$P_i^{nSL} = a * C * V^2 * F_i^{now} \quad (20)$$

$$P_i^{SL} = a * C * V^2 * F_i^{now} + P^{EWMA} + P^{GA} \quad (21)$$

$$P^{GA} = \begin{cases} P^{GA}, & if\ optimisation\ performed \\ 0, & otherwise \end{cases}$$

where P_i^{nSL} and P_i^{SL} is the power consumption in W (watts) in both cases for an IT_i (t seconds), a is the activity factor that represents normalised utilisation of C that is exercised (utilisation) and this is usually close to "1" for Slowing Link, because slowing results in higher utilisation. C is the capacitance that is charged and discharged per clock cycle in Farad. This is equal to $\frac{\text{Watt sec}}{\text{V}^2}$ and C was taken to be 25 nF. V is the supply voltage in Volt, F_i^{now} is clock frequency (operating frequency) in Hz and equal to $\frac{1}{sec}$. P^{EWMA} is power overhead that is consumed by the EWMA and P^{GA} is power consumed by GA. Because of the continuous EWMA monitoring and prediction, the P^{EWMA} is consumed for each SI_i. However, the GA optimisation occurs occasionally, not for each SI_i. For the value of P^{GA} and P^{EWMA}, an alternative approach of investigating an extensive range of P^{GA} and P^{EWMA} values between 3 [12][11] and 70000 mW has been taken. Here, some of the power overheads are set to particularly high values to allow us to investigate sensitivity of our approaches for heavy loading.

Moreover, in order to calculate energy saving for $IT_{1 \to i}$, firstly we calculate the total energy consumption of non-Slowing Link, E^{nSL}, and Slowing Link, E^{SL}, up to the current SI_i. For an entire simulation duration, T^{sim} , we calculate E^{nSL} and E^{SL} Joules (Watt seconds) using the following equations:

$$E^{nSL} = \sum_{i=1}^{T^{sim}} P_i^{nSL} \quad (22) \qquad E^{SL} = \sum_{i=1}^{T^{sim}} P_i^{SL} \quad (23)$$

Finally, in the simulation, the saving is calculated in percent. Two savings were calculated: Instantaneous saving $Save_i$, at i^{th} sample, and total saving $Save^{tot}$ for an entire simulation:

$$Save_i = \left(1 - \frac{P_i^{SL}}{P_i^{nSL}}\right) * 100 \quad (24)$$

$$Save_{1 \to T^{sim}}^{tot} = \left(1 - \frac{E^{SL}}{E^{nSL}}\right) * 100 \quad (25)$$

Figure 8 shows the results of the sample simulation run that performs the calculations/steps mentioned in the previous sections. MATLAB code is written to plot a dashboard that presents the SM parameters on-the-fly for every SI. Researchers are able to view any changes in real-time. The plot presents the InT, Z_t and current operational rate OPR^{now} in the first subplot, current voltage V^{now} and current normalised frequency F^{now} in the second subplot, current average delay D^{hop} in the third subplot, instantaneous power saving $Save_i$ and total power saving $Save^{tot}$ in the fourth subplot and number of alert N^{alert} in the fifth subplot at each SI.

Incoming traffic is shown in the first subplot that is a modified real word router traces and in order to view ability of SM non-realistic extreme bursts are integrated in to the InT. Initially, the SM sets the OPR lower to save power. Bottom left subplot shows instantaneous and total power savings of the frequency reduction. However, around 160$^{\text{th}}$ sample the system produces an alert and SM sets OPR to higher. After observation interval and GA optimisation, the SM decides it is risky (would not satisfy QoS performance requirements) to lower OPR due to extreme bursts and the SM is not lowered the OPR until 300$^{\text{th}}$ sample (second).

At the end of the 300$^{\text{th}}$ sample of exemplar run, the SM establish 10% saving for a non-realistic extreme traffic pattern with a maximum 62% instantaneous power saving.

4. RELATED WORK

The first introduction papers regarding link rate reduction was "Adaptive Link Rate" (slowing) proposed by Gunaratne and Christensen et al. [9][31]. Nedevschi et al. [10] investigated and compared opportunistic sleeping and slowing techniques for network components with consideration of the component's power characteristics. Mandviwalla et al. [8] worked on power consumption of high capacity routers and they presented a simple Dynamic Voltage Scaling (DVS) scheme for an energy-efficient process for multiprocessor-based line cards. One of the earlier hardware based slowing mechanisms for interconnection links was performed by Li Shang et al. [11] and presented a mechanism that observes utilisation of chip-to-chip interconnection and regulates the power consumption by using DVFS techniques to limit throughput. Jung et al. [12] proposed a packet interface power management architecture for reducing energy consumption at the gigabit Ethernet controller. Bolla et al. explored in [32] possibilities of using power management techniques to understand and to characterise the tradeoff between power consumption and forwarding performance. This work provided a general insight as to how an application will be affected by frequency reduction.

5. CONCLUSION AND FUTURE WORK

This paper gathers the building blocks for deploying the power saver slowing mechanism on power hungry communication links. This provides an insight into how a power saving technique can be employed in a network environment and how the parameters of hardware and ICT applications are interlinked with each other. Simulating hardware characteristics of the routers acts as if it is a Device Under Test (DUT) that is capable of providing the SM. The simulation provides flexibility for us and creates InT, packet processing, buffer occupancy, packet delay, throughput, and outputs power consumption and power overhead of the SM.

The feasibility of the SM was theoretically evaluated using the MATLAB simulation and up-to-date equations and a realistic methodology. As a future work, the practicality of the SM for real world scenario can be verified using real testbed facility with a prototype device that performs the SM tasks and establish power saver network as a whole, rather than a single link [33].

6. ACKNOWLEDGEMENT

The authors wish to acknowledge the funding that has been received to support this research project from the EPSRC-BT CASE Award Programme and India-UK Advanced Technology Centre of Excellence in Next Generation Networks, Systems and Services.

7. REFERENCES

[1] K. Kawamoto, J. Koomey, B. Nordman, R. Brown, M. Piette, M. Ting, A. Meier, Electricity used by office equipment and network equipment in the US: Detailed report and appendices, Technical Report LBNL 45917, Lawrence Berkeley National Laboratory, February 2001.

[2] Cisco, The exabyte era, white paper, Jan. 2008.

[3] Cisco, Cisco Visual Networking Index – Forecast and Methodology, 2007-2012, white paper, 16 June 2008.

[4] J. Chabarek, C. Estan, J. Sommers, P. Barford, D. Tsiang, S. Wright, Power Awareness in Network Design and Routing, Proc. IEEE 27th IEEE Conf. on Computer Communications (INFOCOM 2008), Phoenix, AZ, pp. 457- 465, April 2008.

[5] H. Abaci, G. Parr, S. McClean, A. Moore, L. Krug and L. Forgiel, Practical Energy Saving and Power-Workload Proportionality in a campus Environment, in Green Technologies Conference 2012 IEEE (GTC2012), Oklahoma, USA, pp. 1-6, April 2012.

[6] R. Bolla, R. Bruschi, K. Christensen, F. Cucchietti, F. Davoli, and S. Singh, The Potential Impact of Green Technologies in Next Generation Wireline Networks – Is There Room for Energy Savings Optimization?, in IEEE Communication Magazine, vol.49, no.8, pp.80-86, August 2011.

[7] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures, IEEE Commun. Surveys and Tutorials, vol. 13, no. 2, 2nd qtr., pp. 233–244, 2011.

[8] M. Mandviwalla and N.-F. Tzeng, Energy-efficient scheme for multiprocessor-based router linecards, in Proc. SAINT, 2006, pp. 156–163.

[9] C. Gunaratne, K. Christensen, Ethernet Adaptive Link Rate: System Design and Performance Evaluation, Proc. 31st IEEE Conf. Local Computer Networks (LCN 2006), Tampa, FL, USA, pp. 28-35, Nov. 2006.

[10] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, Reducing Network Energy Consumption via Sleeping and Rate-Adaptation, in Proceedings of USENIX NSDI, pp. 323-336, 2008.

[11] L. Shang, L. Peh, and N. K. Jha, Dynamic voltage scaling with links for power optimization of interconnection networks, in Proc. Int. Symp. High-Performance Comput. Architecture, Jan. 2003, pp. 91–102.

[12] H. Jung, A. Hwang, and M. Pedram, Predictive-flow-queue-based energy optimization for gigabit ethernet controllers, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 17, pp. 1113-1126, 2009.

[13] P. Charongrattanasakul and A. Pongpullponsak, Minimizing the cost of integrated systems approach to process control and maintenance model by EWMA control chart using genetic algorithm, in Expert Systems with Applications, Vol. 38, Issue 5, pp. 5178-5186, May 2011.

[14] D.C. Montgomery, G.C. Runger, Applied Statistics and Probability for Engineers. New York: Wiley, 2007

[15] S. Alvarez, QoS for IP/MPLS Networks, Cisco Press, June 2006.

[16] K. Papagiannaki, Provisioning IP Backbone Networks Based on Measurements, Ph.D. Thesis, University College London, March 2003.

[17] D. Miras, A Survey of Network QoS Needs of Advanced Internet Application, Internet2 QoS Working Group, November 2002. http://www.internet2.edu/qos/wg/apps/fellowship/Docs/Internet2AppsQoSNeeds.pdf [Last Accessed September 2009].

[18] M. Mu, A. Mauthe, and F. Garcia, A Utility-based QoS Model for Emerging Multimedia Applications, in First IEEE Future Multimedia Networking (FMN 08) Workshop, Cardiff, UK, pp.521-528, 2008.

[19] J. Evans and C. Filsfils, Deploying IP and MPLS QoS for multiservice networks: theory and practice. Morgan Kaufmann, 2007.

[20] P. Van Mieghem, Performance Analysis of Communication Systems and Networks, Cambridge, U.K.: Cambridge Univ. Press, 2006.

[21] J. Howard, S. Dighe, S.R. Vangal, G. Ruhl, N. Borkar, S. Jain, V. Erraguntla, M. Konow, M. Riepen, M. Gries, G. Droege, T. Lund-Larsen, S. Steibl, S. Borkar, V.K. De and R. Van Der Wijngaart, A 48-Core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling, IEEE Journal Solid-State Circuits, vol. 46, no. 1, pp. 173–183, Jan. 2011.

[22] P. Salihundam, S. Jain, T. Jacob, S. Kumar, V. Erraguntla, Y. Hoskote, S. Vangal, G. Ruhl and N. Borkar, A 2 Tb/s 6 4 Mesh Network for a Single-Chip Cloud Computer With DVFS in 45 nm CMOS, Solid-State Circuits, IEEE Journal, vol. 46, no. 4, pp. 757, April 2011.

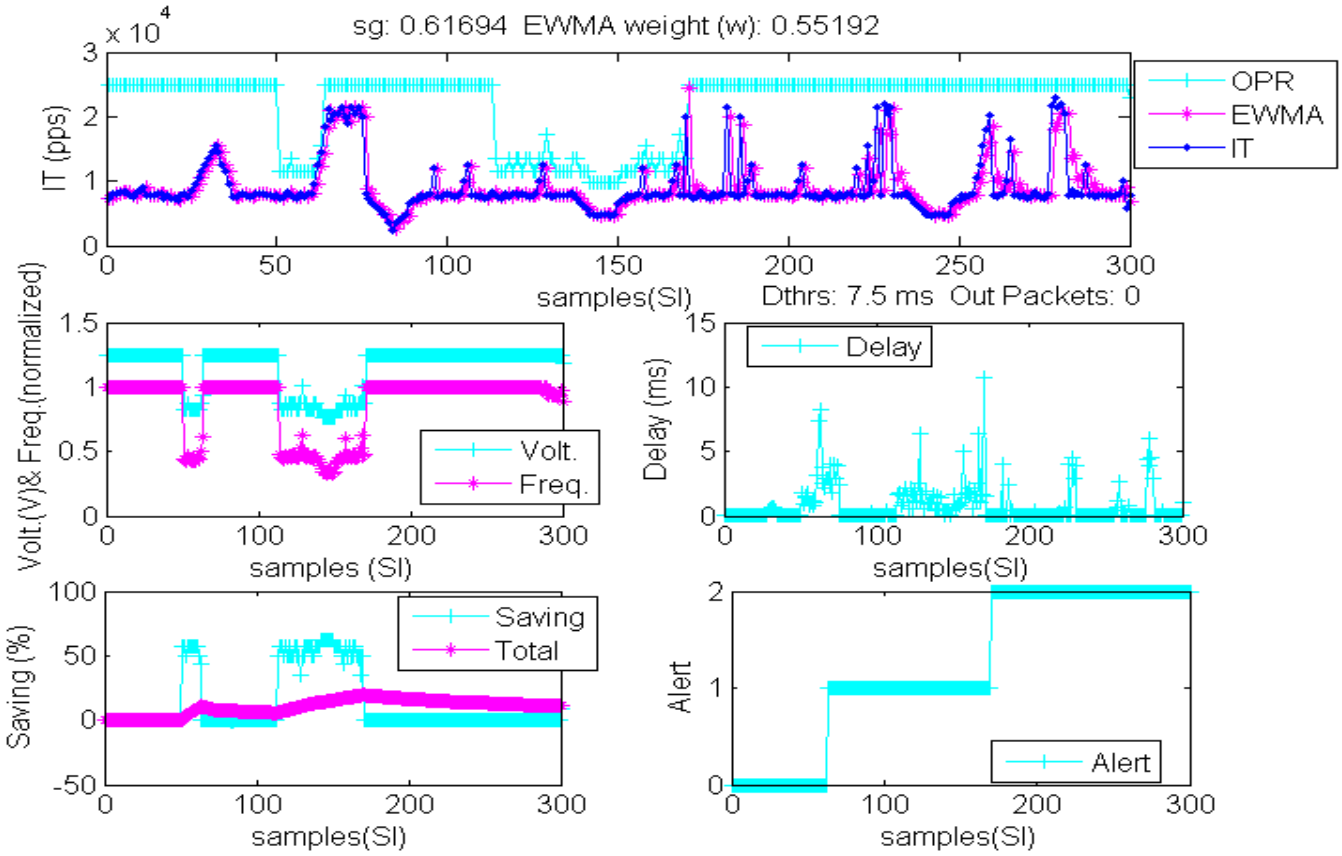

Figure 8. Exemplar snapshot of the SM's dashboard for *SI* between 1 and 300.

[23] D. Medhi and K. Ramasamy, Network Routing: Algorithms, Protocols, and Architectures. Morgan Kaufmann Publishers, 2007.

[24] P. Reviriego, B. Huiszoon, V. López, R.B. Coenen, J.A. Hernández, J.A. Maestro, Improving Energy Efficiency in IEEE 802.3ba High-Rate Ethernet Optical Links, Selected Topics in Quantum Electronics, IEEE Journal, 17(2), 419 – 427, April 2011.

[25] R. Hays, Active/Idle Toggling with 0BASE-x for Energy Efficient Ethernet, presentation to the IEEE 802.3az Task Force, Nov. 2007 http://www.ieee802.org/3/az/public/nov07/hays 1 1107.pdf

[26] E. Macii (Ed), Ultra Low-Power Electronics and Design. Kluwer Academic Publishers, 2004.

[27] Cisco, CRS-1, Datasheets (2011), http://www.cisco.com/en/US/prod/ collateral/routers/ps5763/prod_brochure0900aecd800f8118.pdf [Last Accessed June 2012]

[28] N. Beheshti, Y. Ganjali, M. Ghobadi, N. McKeown, and G. Salmon, Experimental study of router buffer sizing, in Proc. IMC, Vouliagmeni, Greece, pp. 197–210, Oct. 2008.

[29] M. Keating, D. Flynn, R. Aitken, A. Gibbons, K. Shi, Low Power Methodology Manual for System on Chip Design. Springer-Verlag, 2007.

[30] E. Gelenbe, A Software Defined Self-Aware Network: The Cognitive Packet Network, Network Cloud Computing and Applications (NCCA), 2014 IEEE 3rd Symposium on , vol., no., pp.9,14, 5-7 Feb. 2014.

[31] C. Gunaratne, K. Christensen, B. Nordman and S. Suen, Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR), IEEE Transactions on Computers, vol. 57, issue 4, April 2008 pp.448-461.

[32] R. Bolla, R. Bruschi, and A. Ranieri, Green support for PC-based software router: Performance evaluation and modeling, in Proc. IEEE ICC, Dresden, Germany, pp. 1–6, Jun. 2009.

[33] E. Gelenbe and C. Morfopoulou. A Framework for Energy Aware Routing in Packet Networks. The Computer Journal, 54(6), June 2011.