

On-line Power Savings in a Distributed Multi-stage Router Architecture

Andrea Bianco, Fikru Getachew Debele, Luca Giraudo

Dipartimento di Elettronica e delle Telecomunicazioni, Politecnico di Torino, Italy

Email: andrea.bianco@polito.it, {fgetachew, luca.giraudo}@gmail.com

Abstract— We focus on a distributed multi-stage software router (MSSR) architecture internally composed by several personal computers (PCs) to overcome scalability and performance issues of software routers (SRs) based on a single PC. Sizing the internal architecture to sustain the peak load may lead to power inefficiency at low loads. This paper presents a power saving scheme to improve the power efficiency of the MSSR by dynamically adapting the size of its internal architecture to the offered load to reduce power needs. The off-line problem is defined as a mixed integer linear programming optimization model, shown to be NP-hard. We propose a differential on-line heuristic to solve the optimization problem when the traffic load changes. The heuristic avoids the complete MSSR reconfiguration of the optimal off-line solution that may lead to forwarding delay increase or service interruption. The performance evaluation shows that the proposed on-line algorithm, that gracefully modifies the internal MSSR configuration, preserves the load proportional power demand characteristics of the optimal off-line solution.

I. INTRODUCTION

The research community has devoted a lot of attention to software routers (SRs), routers based on personal computers (PCs) running open-source network application software like Linux, Click Modular Router [1], Quagga [2] or XORP [3]. The main benefits of SRs include the wide availability of multi-vendor hardware and documentation, low device cost, and the continuous performance evolution driven by the PC market economy of scale. Furthermore, open source SRs provide the opportunity to easily modify the router operation, resulting in flexible and configurable routers, whereas proprietary network devices often lack programmability and flexibility, have high costs and require complex management procedures in a multi-vendor scenario. Criticisms to PC-based SRs are focused on limited performance, software instability, lack of system support, scalability issues, and lack of advanced functionalities. To overcome some of these limitations, a distributed multi-stage software router (MSSR) architecture, shown in Fig. 1, that exploits classical PCs as elementary switching elements to build high-performance SRs was proposed in [4]. This architecture permits to: i) overcome performance limitations of SRs based on a single PC by offering multiple, parallel forwarding paths; ii) gracefully increase SR performance by incrementally adding/upgrading internal elements; iii) scale the number of interfaces; and iv) enhance faults resilience.

The proposed architecture has three stages: i) the layer-2, open-software or open-hardware based [5], load balancers

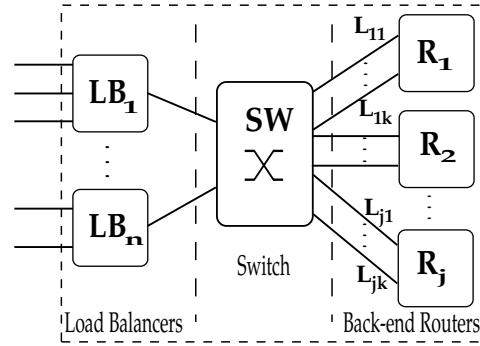


Fig. 1. MSSR Architecture: the load balancers (LB) (first stage), the switch (second stage) and the back-end routers (third stage)

(LBs) act as interfaces to the external networks and distribute IP packets to ii) back-end PCs, also named *back-end routers* in the paper, that provide IP routing functionality, through iii) an interconnection network, based on Ethernet switches, that connects the two stages. A control entity, named *Virtual Control Processor* (VirtualCP), which runs on a selected back-end router, controls and manages the overall architecture through the DIST protocol [6]. The VirtualCP hides the internal details of the MSSR architecture to external network devices.

Like many networking devices, the MSSR is typically sized for peak traffic. State-of-the-art PC-based routers can route few Gbps if the packet processing is performed by the CPU [7] [8] or few tens of Gbps if a specialized packet processing is implemented [9]. Therefore, the MSSR architecture might require tens of back-end PCs to achieve high-end performance. This performance scaling implies a high redundancy level at the back-end stage, which leads to power wastage at low loads. Instead, during low traffic periods, the routing task could be transferred to a subset of back-end PCs setting all other back-end PCs in off state to save power.

Let us discuss this issue using a realistic scenario. Suppose the goal is to design a MSSR equivalent to a Juniper T320 core router that supports up to sixteen 10 Gbps ports and has 160 Gbps forwarding capacity [10]. The following internal components are available to design the MSSR:

- back-end PCs with 5.5 Gbps forwarding capacity and equipped with a single 10 Gbps interface;
- LBs with two (one internal and one external) 10 Gbps interfaces;

- one (or a set of) hardware Ethernet switch(es) with enough capacity to interconnect LBs and back-end routers.

To design a 160 Gbps capable MSSR, 16 LBs and 29 back-end routers are needed. If we assume for the sake of simplicity both LBs and back-end routers running on PCs with idle power consumption equal to 80 Watts (a reasonable power consumption for today computers), this architecture dissipates 3.6 kW in idle state, excluding the power of the interconnecting switch(es). However, in idle state or during low traffic periods, one (or few) back-end PCs may be enough to give the required service. Thus, it would be possible to reduce the MSSR consumption to 1.36 kW in idle state, the power consumption of 16 LBs and 1 back-end PC. In other words, the input traffic can be consolidated to few back-end PCs setting all other PCs to low power state to save power. Power saving for LBs and the switch(es) is not considered in the paper, because they act respectively as external interfaces (which must stay active to guarantee external connectivity) and internal interconnection network (which must be active to guarantee internal connectivity). As such, saving power by switching off LBs is only possible when operating at the network level, as in [11], where the whole network power consumption is optimized by redirecting the traffic over a subset of routers.

Therefore, we propose an optimization algorithm that resizes the number of back-end PCs, characterized by different power consumption and routing capacity, to adapt the MSSR overall capacity to the incoming traffic demand so as to minimize the power consumption. The remainder of the paper is organized as follows: in Sec. II we discuss previous research on power savings in networks and devices. In Sec. III we describe the off-line power saving problem in the MSSR architecture and give a detailed formulation of the problem. The proposed on-line optimization algorithm is described in details in Sec. IV, whereas Sec. V discusses performance results. Finally, we conclude the paper in Sec. VI.

II. RELATED WORK

Rising power cost and increasing environmental standards urge researchers and industries to draw attention to power reduction aspects of data networking. While power efficiency has been a major technology driver in the mobile and embedded area for some time, the power saving in data network has been only recently addressed. Starting from a position paper by Gupta et al. [12], researchers in IT focused on how to save power in data network. Bolla et al. gives a detailed survey [13] on emerging technologies, projects, and work-in-progress standards which can be adopted in networks and related infrastructures to reduce their power requirements and carbon footprint.

Chase et al. [14] proposed a power-conscious user request switching paradigm to reduce power consumption for server cluster during low load periods. The switch monitors cluster load and concentrates traffic on the minimal set of servers that can serve user requests with given utilization and latency

constraints. This induces the remaining idle servers to step down to a low-power state. The proposal basically extends the load-balancing switches with a power-conscious routing policy that leverages the power management features of the back-end servers. The scheme permits power saving only at the coarse granularity of the entire server. Furthermore, only homogeneous servers are considered. In our MSSR architecture, PCs acting as back-end routers are heterogeneous, both in capacity and power consumption, and they may be equipped with several network cards. Thus, we are not limited to operate on switching on and off PCs but we can also act at the card level, taking into account that a 10 Gbps link consumes roughly 20 W [15], about one fourth of a standard PC consumption.

Two main forms of power saving in networks, rate adaptation and sleeping of network devices, have been proposed [16]. Researchers exploit these options to save power in a network through smart topology reconfiguration. Chiaraviglio et al. [11] addressed a network design problem by considering the minimization of the total power consumed by the network. They proposed heuristics to select a minimum set of routers and links to be used to support a given traffic demand. The main idea is to power off links and even full routers while guaranteeing QoS constraints such as maximum link utilization. A novel power reduction approach at the network level by considering nodes capable of adapting their performance to the actual load has also been proposed in [17]. However, those researchers focus on power-aware routing, whereas we focus on power-conscious traffic aware router dimensioning.

An off-line MSSR power saving scheme has been proposed in [18]. Since the optimal problem is not solvable for large scale routers, the authors proposed a two-step heuristic approach to split the problem into links and routers optimization problem. The results show that the two-step algorithm scales to a large size MSSR and its solution is within 10% relative error of the optimal solution. In this paper we focus on an on-line solution following a differential approach, i.e., the MSSR is not fully reconfigured but the new configuration, needed to adapt to the modified traffic load, is obtained by resizing the current one either activating or de-activating PCs, without creating any service interruption. We compare the performance of our on-line algorithm with respect to the optimal off-line solution.

III. SYSTEM MODEL

We design a mechanism to reduce power dissipation in the considered MSSR architecture by adapting the number of back-end PCs to the currently offered traffic load. The problem can be stated as follows: **Given** i) a set of back-end PCs B ; each PC $r \in B$ characterized by power consumption (excluding network cards) $P_r \in \mathbb{R}$ and routing capacity $C_r \in \mathbb{R}$, ii) a set of links L_r connected to each PC $r \in B$; each link $l \in L_r$ characterized by power consumption $P_{rl} \in \mathbb{R}$ and link capacity $C_{rl} \in \mathbb{R}$, and iii) an aggregate input traffic demand $T \in \mathbb{R}$, **Select** PCs and links required to route the traffic demand such that the power consumption of the

MSSR architecture is minimized, **Subject to** link rate and PC capacity.

The problem formulation is based on the following assumptions:

- 1) *The input traffic T is splittable among back-end PCs:* every packet is managed independently by LBs, and each LB is responsible to load balance the incoming traffic among active back-end PCs. The aggregate incoming traffic T is measured and made available to the Virtual CP which runs the optimization algorithm.
- 2) *PCs and NICs power consumption:* optimization of PCs and NICs power consumption are done separately. We consider a single link per card scenario. Therefore, we represent the combined power consumption of a card and its link l on a given PC r by $P_{rl} \in \mathbb{R}$. Hence, the maximum power consumption of a back-end PC r is given by $P_r + \sum_l P_{rl}$.
- 3) *ON-OFF power model:* to keep the problem formulation simple, we chose the ON-OFF power model [17] both for back-end PCs and links; i.e., the power consumption does not depend on the actual resource load, but it is either zero when the resource is off or equal to a fixed value when the resource is on.

Given the above definitions and assumptions, the MSSR power saving scheme can be formalized as a mixed integer linear programming (MILP) problem. Let α_r be a PC selection binary variable (equal to 1 if the PC $r \in B$ is activated, 0 otherwise), β_{rl} the link selection binary variable (equal to 1 if the link $l \in L_r$ connected to PC $r \in B$ is used, 0 otherwise), t_{rl} the portion of traffic T to be forwarded by PC r on link l . Thus, the problem is formulated as:

minimize

$$P = \sum_r (P_r \alpha_r + \sum_l P_{rl} \beta_{rl}) \quad (1)$$

subject to

$$\sum_r \sum_l t_{rl} = 1 \quad (2)$$

$$\sum_l t_{rl} T \leq C_r \alpha_r \quad \forall r \in B \quad (3)$$

$$t_{rl} T \leq C_{rl} \beta_{rl} \quad \forall r \in B, \forall l \in L_r \quad (4)$$

$$\alpha_r \geq \beta_{rl} \quad \forall r \in B, \forall l \in L_r \quad (5)$$

$$\alpha_r, \beta_{rl} \in \{0, 1\}, t_{rl} \in [0, 1] \quad (6)$$

In the MILP formulation, (2) ensures that all the input traffic T is served, while (3) and (4) ensure that the capacity constraints of each PC (C_r) and link (C_{rl}) are not violated. (5) guarantees that PC r is active if at least one of its links is chosen to transport some traffic.

Equations (1) – (6) define a MILP problem that optimizes the MSSR architecture power consumption, considering both PCs and NICs simultaneously. The problem is NP-hard as demonstrated in [19]. Thus, exact methods can only be used to solve small size cases. Furthermore, in a variable traffic demand scenario, when a new MSSR configuration must be defined, the PCs selection may be completely different from the previous one. In this paper we propose a simple

differential on-line heuristic, i.e., an algorithm that computes the new MSSR configuration taking into account the new traffic demand as well as the current MSSR configuration.

IV. PROPOSED HEURISTIC ALGORITHM

The proposed power saving heuristic is an on-line differential algorithm that defines the new back-end stage configuration needed to satisfy the current traffic demand by modifying the existing MSSR configuration. Let us first describe it considering its operation when no current MSSR configuration has been designed, i.e., when the algorithm is run for the first time. All PCs are assumed to be in the off state.

The power saving algorithm works as follows: At start up, the MSSR available configuration, i.e. the set of PCs and of their NIC cards available for the back-end stage, is analyzed to determine the actual maximum routing available capacity (C_r) of each PC and of the whole architecture. These values are the capacity constraints presented in the MILP formulation. We assume that the maximum available MSSR architecture capacity is enough to deal with the traffic capacity made available to external networks by the LBs. The actual routing capacity of a PC acting as a router is limited either by the CPU packet processing capacity or by the sum of link rates on the PC. The algorithm considers the minimum between these two capacities as the actual routing capacity. The CPU packet processing capacity of PCs used as routers is reported, for example, in [7] [8].

The goal of the algorithm is to activate, i.e., to set in the on state, the less power hungry subset of PCs as back-end routers able to manage the requested traffic demand. Two slight variations are considered. The version of the algorithm denoted as NIC- does not consider the NIC power consumption when evaluating PCs' efficiency. More precisely, the algorithm sorts PCs according to their efficiency η_{NIC-} , defined as the amount of traffic routed per watt:

$$\eta_{NIC-} = \frac{C_r}{P_r} \quad (7)$$

Instead, the version of the algorithm that takes into account also the link power consumption during PCs sorting stage is named NIC+. In this case, Eq. 7 can be rewritten as:

$$\eta_{NIC+} = \frac{C_r}{P_r + \sum_{i=1}^N P_{rl}} \quad (8)$$

where N is the number of links connected to back-end PC r .

Similarly, links in a PC r are also sorted according to their efficiency η_l :

$$\eta_l = \frac{C_{rl}}{P_{rl}} \quad (9)$$

After the PC sorting phase, the algorithm follows a greedy approach. It starts activating (i.e., setting in the on state) the available most efficient PC r according to Eq. 7 or 8 using the available most efficient links l on PC r according to Eq. 9. More precisely, the algorithm tries to fully exploit the most efficient PC with a portion of the incoming traffic equal to the PC actual capacity. Links are activated to satisfy capacity

requirements and are selected in order of their efficiency. When the first PC is fully utilized, if a residual traffic has to be served, the algorithm considers packing the residual incoming traffic to the next available most efficient PC. This procedure is iterated until all the incoming traffic is served. If the incoming traffic exceeds the MSSR architecture capacity, the extra amount of traffic is discarded.

After this initial design procedure, the algorithm monitors the incoming traffic demand to identify a traffic modification worth of a reconfiguration phase. How frequently the monitoring procedure should run and how the algorithm determines when to reconfigure the back-end stage are issues not considered in this paper. Classical measurement algorithms and threshold based activation scheme could be used to solve this problem.

When a re-configuration request is triggered, if the traffic demand increases, the algorithm computes the extra traffic demand and turns ON additional resources, i.e., links on already active PCs and new PCs currently in off state, needed to route the increased demand. As in the previous case, links and PCs are considered in order of their efficiency. If the traffic decreases, the algorithm adjusts the running configuration by turning off links and PCs, again considering them in order of their efficiency, i.e. less efficient links and PCs first.

The algorithm is a modified first-fit-decreasing algorithm [20] with bins (PCs) having different size and different usage cost (power consumption). Furthermore, the power saving algorithm must also consider bins (links) within a bin (PC).

Fig. 2 depicts a block diagram of the proposed power saving scheme running in the MSSR architecture. The power saving scheme is assumed to run on the same back-end router where the VirtualCP is running to ease their interaction. The VirtualCP monitors, through the DIST protocol, any change in the MSSR configuration, to identify any modification in link and PCs configuration, that may be caused by device faults, upgrade or addition/removal for management purposes. These modifications trigger create/update actions on the system model instance. The traffic statistic module collect traffic information and updates the system model instance.

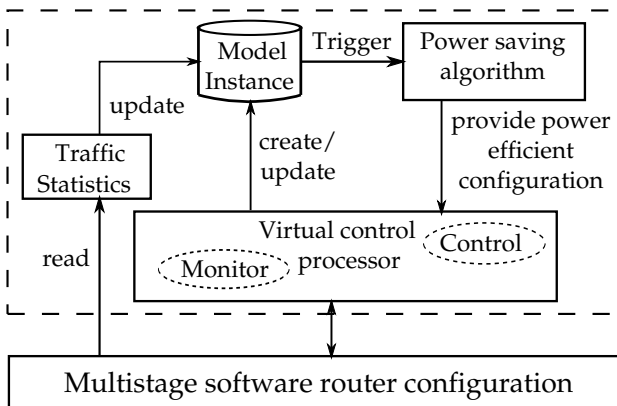


Fig. 2. MSSR architecture power saving scheme block diagram

Any change in the input traffic above a predefined threshold or any modification in PC configuration triggers the power saving algorithm that defines a new power efficient MSSR configuration.

Once the new MSSR configuration has been defined by the power saving algorithm, the VirtualCP switches on and off the proper set of PCs and links exploiting the DIST protocol features. Furthermore, load balancing tables of front-end stage LBs are modified accordingly, to ensure that the incoming traffic is forwarded only to currently active PCs.

V. PERFORMANCE EVALUATION

To assess the performance of the proposed algorithm, we consider a MSSR architecture in which three different groups of PCs are available as back-end routers. Each group consists of five PCs and each PC in each group has the following specification:

Group I

- Back-end router routing capacity $C_r = 4000$ Mbps [7];
- PC power consumption $P_r = 60$ W [21];
- Link capacity $C_{rl} = 1000$ Mbps (4 links per PC);
- Link power consumption $P_{rl} = 4$ W [15];

Group II

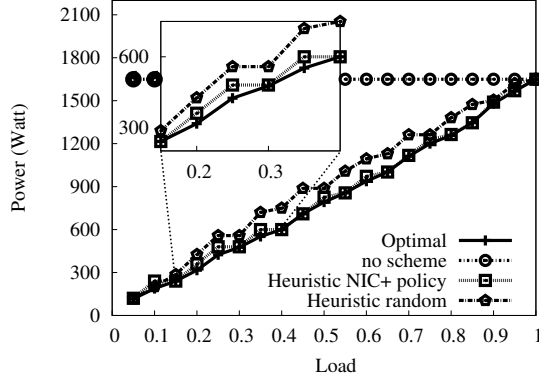
- Back-end router routing capacity $C_r = 8700$ Mbps [8];
- PC power consumption $P_r = 100$ W;
- Link capacity $C_{rl} = 10000$ Mbps (1 link per PC);
- Link power consumption $P_{rl} = 20$ W [15];

Group III

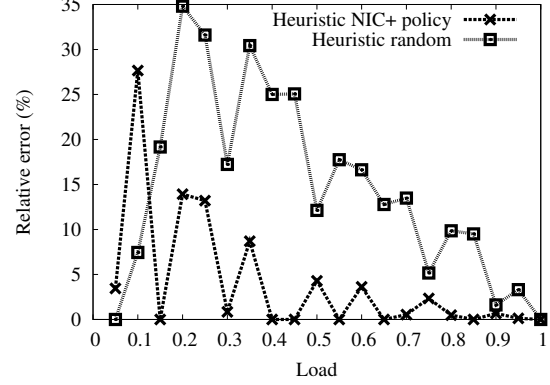
- Back-end router routing capacity $C_r = 8700$ Mbps;
- PC power consumption $P_r = 80$ W;
- Link capacity $C_{rl} = 1000$ Mbps (9 links per PC);
- Link power consumption $P_{rl} = 6$ W;

For the second and third groups the actual routing capacity is limited by capacity of the PCs because the total link capacity of each PC is larger than its routing capacity. Instead, for the first group the router capacity and the total link capacity are the same. Overall, the multi-stage software router has a maximum routing capacity of 107 Gbps. To fully utilize this capacity, we assume 11 LBs each with two (one internal and one external) 10 Gbps link. Assuming 80 W of power consumption for each LB, without any power saving scheme the architecture consumes 2.53 kW (1.65 kW by the back-end routers, as shown in Fig. 3(a) with the curve labelled no scheme) excluding the interconnecting switch.

To demonstrate the power saving that can be achieved by running the heuristic, we compare the on-line heuristic with the optimal off-line algorithm, and with an on-line heuristic version that is not sorting PCs (labelled random in the plots), assuming a traffic load increasing from 0.1 to 1, where 1 represents the normalized maximum MSSR capacity. The traffic load increase step is set to 0.05, i.e., the on-line heuristic is run for changes in traffic load larger than 0.05. The optimal MILP problem is solved for every traffic load on CPLEX [22]. We consider the two router sorting policies NIC+ and NIC- to highlight the impact of link power constraint when sorting



(a) Power saving achieved by different schemes



(b) Heuristic relative error

Fig. 3. MSSR architecture power saving schemes comparison (NIC+ sorting policy)

PCs. Recall that no power savings on LBs are considered, as discussed in Sec. I. Therefore, the results presented include only the power consumption of PCs acting as back-end routers.

Fig. 3(a) compares the power saving that can be achieved by the NIC+ policy. The small inset in Fig. 3(a) magnifies the power consumption of the three algorithms in a load range between 0.15 to 0.4. A large power saving is obtained at low loads by the heuristic, up to 1.53kW, with respect to the curve labelled no scheme that refers to the maximum capacity MSSR configuration. Fig. 3(b) reports also the relative error of the proposed on-line NIC+ and of the random heuristic with respect to the optimal off-line. The NIC+ heuristic approximates fairly well the optimal off-line solution, with a worst case error of about 30%. The sorting algorithms enhance significantly the heuristic performance, as shown by the worse performance of the random heuristic.

Although the power savings of the off-line optimal approach are larger, the on-line heuristic has an obvious scalability benefits. Furthermore, it gracefully modifies the current MSSR configuration minimizing service interruptions. Fig. 4 supports the claim that the on-line heuristic is less disruptive compared to the off-line optimal reconfiguration. The plot reports the difference in the number of PCs between two consecutive

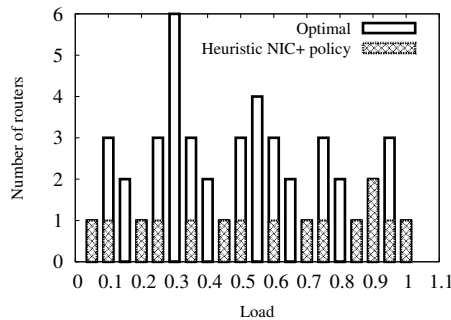


Fig. 4. Configuration difference between two consecutive solution

configurations for the optimal off-line and the heuristic on-line algorithm. The heuristic requires the activation of a minimal number of PCs to follow traffic load increase. For instance, a traffic load transition from 0.25 to 0.3 results in turning on and/or off 6 PCs in the case of the optimal solution whereas no activation of PCs is required for the heuristic solution.

Finally, Fig. 5 highlights the importance of taking into account also link power in sorting back-end PCs. Similar observations to those discussed when looking at Fig. 3 hold when comparing the heuristic with the optimal off-line or the random heuristic. Furthermore, we observe that considering link power consumption in sorting PCs improves the heuristic efficiency. A comparison of Figs. 3(b) and 5(b) shows that the NIC+ policy outperforms, on average, the NIC-.

In summary, our main findings are:

- The on-line heuristic makes the power consumption of the back-end stage of the MSSR proportional to the input load ensuring large power savings;
- Although slightly less efficient, the heuristic is largely less disruptive compared to the optimal approach. Furthermore, whereas the optimal solution has some scalability problems when dealing with large size MSSR architecture [18], the simplicity of the on-line heuristic approach ensures that large size MSSR can be managed.
- The NIC+ sorting policy outperforms the NIC- policy on average.
- The PC sorting procedure based on PC power efficiency significantly enhance the saving performance.

VI. CONCLUSIONS

We presented a differential on-line power saving heuristic to adapt the power consumption of a MSSR architecture to the traffic load by properly choosing a set of active PCs in the back-end stage to match the incoming traffic load. We evaluated the on-line algorithm with respect to the previously studied optimal off-line power saving approach.

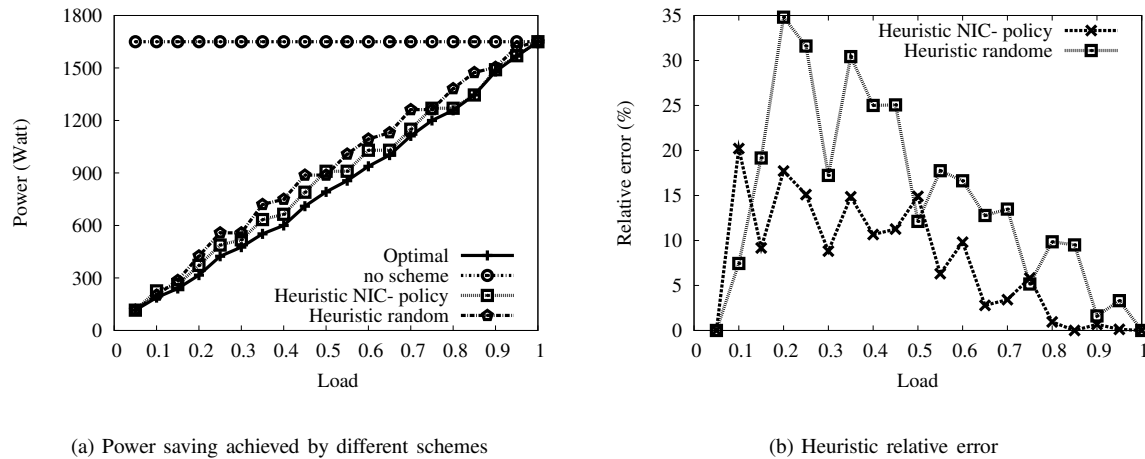


Fig. 5. MSSR architecture power saving schemes comparison (NIC- routers sorting policy)

The results shows that, even though the on-line heuristic solution is not optimal, it provides significant power savings compared to the no saving scheme, with power requirements not far from those of the optimal off-line case. The capability of the on-line heuristic to define the new MSSR configuration as a slight modification of the existing one permits to minimize the potential service interruption that a full reconfiguration, required by the off-line approach, would create. Finally, the simple heuristic approach scales well to large size MSSR architectures.

ACKNOWLEDGMENTS

This research was funded by the Italian Ministry of Research and Education through the PRIN SFINGI (SoFtware routers to Improve Next Generation Internet) project.

REFERENCES

- [1] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click modular router," *ACM Trans. Comput. Syst.*, vol. 18, pp. 263–297, August 2000.
- [2] "Quagga routing suite." [Online]. Available: <http://www.nongnu.org/quagga/> [Accessed: March, 2012]
- [3] M. Handley, O. Hodson, and E. Kohler, "XORP: an open platform for network research," *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 53–57, January 2003.
- [4] A. Bianco, J. Finochietto, M. Mellia, F. Neri, and G. Galante, "Multistage switching architectures for software routers," *Network, IEEE*, vol. 21, no. 4, pp. 15–21, July–August 2007.
- [5] M. Petracca, R. Birke, and A. Bianco, "HERO: High-speed enhanced routing operation in Ethernet NICs for software routers," *Computer Networks*, vol. 53, no. 2, pp. 168–179, 2009.
- [6] A. Bianco, R. Birke, J. Finochietto, L. Giraudo, F. Marenco, M. Mellia, A. Khan, and D. Manjunath, "Control and management plane in a multi-stage software router architecture," in *High Performance Switching and Routing, 2008. HSPR 2008. International Conference on*, May 2008, pp. 235–240.
- [7] A. Bianco, R. Birke, D. Bolognesi, J. Finochietto, G. Galante, M. Mellia, M. Prashant, and F. Neri, "Click vs. Linux: two efficient open-source IP network stacks for software routers," in *High Performance Switching and Routing, 2005. HPSR. 2005 Workshop on*, May 2005, pp. 18–23.
- [8] M. Dobrescu, N. Egi, K. Argyraki, B. Chun, K. Fall, G. Iannaccone, A. Knies, M. Manesh, and S. Ratnasamy, "RouteBricks: Exploiting parallelism to scale software routers," in *ACM SOSP*, 2009, pp. 15–28.
- [9] S. Han, K. Jang, K. Park, and S. Moon, "PacketShader: a GPU-accelerated software router," *SIGCOMM Comput. Commun. Rev.*, vol. 40, pp. 195–206, August 2010.
- [10] "T Series Core Routers." [Online]. Available: www.juniper.net/us/en/local/pdf/datasheets/1000051-en.pdf [Accessed: March, 2012]
- [11] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware backbone networks: A case study," in *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, June '09, pp. 1–5.
- [12] M. Gupta and S. Singh, "Greening of the Internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '03. ACM, 2003, pp. 19–26.
- [13] 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," *Communications Surveys Tutorials, IEEE*, vol. 13, no. 2, pp. 223–244, quarter 2011.
- [14] J. Chase and R. Doyle, "Balance of power: Energy management for server clusters," in *Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS)*, 2001, pp. 163–165.
- [15] Broadcom Corporation, "Broadcom NetXtreme II Network Adapter User Guide." [Online]. Available: http://www.broadcom.com/docs/support/ethernet_nic/Broadcom_NetXtremeII_Server_T7.0.pdf [Accessed: March, 2012]
- [16] S. Nadevski, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI'08. USENIX Association, 2008, pp. 323–336.
- [17] J. Restrepo, C. Gruber, and C. Machuca, "Energy profile aware routing," in *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, June 2009, pp. 1–5.
- [18] A. Bianco, G. Fikru, and L. Giraudo, "Energy saving in distributed router architectures," *Communications (ICC), 2011 IEEE International Conference on*, June 2012.
- [19] —, "Energy saving in distributed router - Tech report." [Online]. Available: http://www.telematica.polito.it/debele/reports/energy_saving_technical_report_2011.pdf [Accessed: March, 2012]
- [20] E. G. Coffman Jr, M. R. Garey, and D. S. Johnson, *Approximation algorithms for bin packing: a survey*. PWS Publishing Co., 1997, pp. 46–93.
- [21] "Approximate desktop, notebook, & netbook power usage." [Online]. Available: <http://www.upenn.edu/computing/provider/docs/hardware/powerusage.html> [Accessed: March, 2012]
- [22] "IBM ILOG CPLEX Optimization Studio." [Online]. Available: <http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/> [Accessed: March, 2012]