Energy saving in distributed router architectures

Energy Saving in Distributed Router Architectures

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— **A multi-stage software router overcomes scalability issues related to a single, PC-based, software router by introducing parallel forwarding paths. However, since the architecture includes different internal components, energy inefficiency at low loads may arise if the multi-stage internal architecture does not adapt to currently offered traffic.**

This paper presents an energy-saving scheme to improve energy efficiency of the multi-stage router architecture by focusing on the back-end stage and sizing it to the offered load to reduce energy needs. The problem is defined as a mixed integer linear programming model, shown to be NP-hard. We tackle the scalability issues of the optimal problem by defining a two-step heuristic which takes advantage of existing BIN PACKING algorithms. Our results shows that the two-step solution is within 10% relative error with respect to the optimal solution for different realistic scenarios.

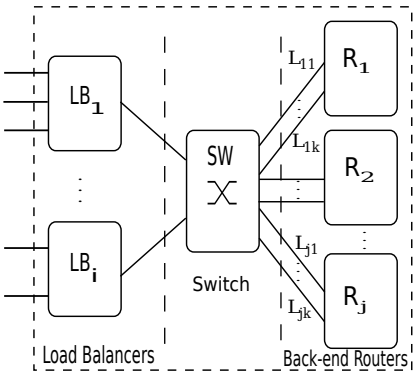


Fig. 1. MSSR Architecture: the load balancers(LB) (first stage), the switch (second stage) and the back-end routers (third stage)

I. INTRODUCTION

Networking equipments, and routers in particular, are characterized by the development of proprietary architectures, often leading to high cost in terms of both equipment and training, because network administrators need to manage different vendor devices or they are forced to a single vendor scenario. This situation drove researchers to identify software routers (SRs) as an appealing alternative to proprietary devices. SRs are based on personal computers (PCs) running open-source network application software like Linux, Click Modular Router [1] or XORP [2].

The main benefits of SRs include: wide availability of multi-vendor hardware and documentation, low cost and continuous 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.

Criticisms to single PC-based SRs are focused on limited performance, software instability, lack of system support, scalability issues, and lack of advanced functionalities. To overcome these limitations, a multi-stage architecture (shown in Fig. 1) that exploits classical PCs as elementary switching elements to build high-performance SRs was proposed in [3]. The key advantages of this architecture are the ability to: i) overcome performance limitations of single PC-based routers by offering multiple, parallel forwarding paths; ii) increase router performance by incrementally adding/upgrading internal elements; iii) scale the number of interfaces; and iv) enhance faults resilience.

The proposed architecture has three stages: the layer-2 front-end load balancers (LBs) act as interfaces to the external networks, the back-end PCs (also named as *back-end routers* or *routers* in the rest of the paper) provide IP routing functionality, and an interconnection network based on Ethernet switches to connect 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 [4]. The VirtualCP hides the internal details of the multi-stage router to external network devices.

Like many networking devices, the multi-stage software router is typically sized for peak traffic. State-of-the-art PC-based routers can route only few Gbps [5], [6] if the packet processing is performed by the CPU or few tens of Gbps if a specialized packet processing is implemented [7]. Therefore, the multi-stage architecture might require tens of back-end routers to achieve high-end performance. This performance scaling implies a high redundancy level at the back-end stage, which translates into a source of energy wastage at low loads. Hence, during low traffic periods, the routing task can be transferred to a subset of back-end routers setting all other back-end routers in low power state to save energy.

While back-end routers are redundant during low traffic periods, LBs and switches are not 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). Thus, saving energy by switching off LBs is only possible when operating at the network level, as in [8], where the whole network energy consumption is optimized by redirecting the traffic over a subset of routers.

In light of these considerations, we propose a mixed integer linear programming model (MILP) to select a subset of back-

end routers, characterized by different power consumption and routing capacity, so as to minimize overall power consumption while satisfying the traffic demand. The problem is a typical resource allocation problem: objects (the traffic) have to be packed in a set of bins (the routers) while minimizing a total cost (the power consumption). Thus, while our research focuses on multi-stage software router, the proposed technique can be applied to many load distribution scenarios, e.g., clusters where a set of resources (task) and a set of containers (processors) with an associated cost and capacity are deployed.

The main paper contributions are the MILP formulation of the energy saving problem, the description of a two-step heuristic to solve the NP-hard model and the performance analysis of the proposed algorithms.

II. Problem definition and MILP formulation

We design mechanisms to reduce energy consumption in the multi-stage SR architecture by adapting the number of back-end routers to the currently offered traffic load. In this section we introduce the notation and variables used to describe the optimization problem as a MILP model.

- **Routers**: B is the set of back-end routers. $\forall r \in B$, $P_r \in \mathbb{R}$ is the router power consumption (excluding line cards) and $C_r \in \mathbb{R}$ is the routing capacity
- **Links**: L is the set of all internal links. $L_r \subseteq L$ is the set of internal links connected to router r. $\forall r \in B$, $\forall l \in L_r$, $P_{rl} \in \mathbb{R}$ is the link power consumption and $C_{rl} \in \mathbb{R}$ is the link capacity
- **MILP variables**: $\forall r \in B$, α_r is a router selection binary variable (equal to 1 if the router is activated, 0 otherwise). $\forall r \in B, \forall l \in L_r$, β_{rl} is the link selection binary variable (equal to 1 if the link is used, 0 otherwise). The total input traffic is denoted by $T \in \mathbb{R}$, whereas t_{rl} is the portion of traffic T to be forwarded by router r on link l.

The problem formulation is based on the following assumptions:

1) *The input traffic T is splittable among the back-end routers*: every packet is managed independently by LBs. Each LB is responsible to load balance the incoming traffic among active back-end routers. The aggregate incoming traffic T is measured and available to the Virtual CP which runs configuration optimization.
2) *Routers and NICs energy consumption*: optimization of routers and NICs energy consumption is done separately. In this work we consider a single link per card scenario. However, a back-end router can be connected to the interconnecting switch with more than one NIC. Therefore, we represent the combined power consumption of a card and its link l on a given router r by $P_{rl} \in \mathbb{R}$. Hence, the maximum power consumption of a back-end router r is given by $P_r + \sum_l P_{rl}$.
3) *ON-OFF power model for the back-end routers and links*: to keep the problem formulation simple, we chose the ON-OFF energy model both for the routers and the links; i.e., the energy consumption does not depend on the actual resource load, but it is either zero when the resource is off or equal to a constant value.
4) *Off-line solution*: the energy saving problem is solved using an *off-line*, memoryless algorithm: The algorithm is not taking into account the evolution of the input traffic, but it is designed to give the best solution given a specific input traffic.

Thus, the multi-stage software router energy saving scheme can be formalized as a MILP problem as follows:

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

$$\text{s.t.} \quad \sum_r \sum_l t_{rl} = 1 \tag{2}$$

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

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

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

$$\alpha_r, \beta_{rl} \in \{0,1\}, t_{rl} \in [0,1] \tag{6}$$

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

Equations (1)–(6) define a MILP problem that optimizes the multi-stage architecture power consumption, considering both routers and NICs simultaneously. Hence, we refer to it as the *combined problem* in the next sections. The problem can be shown to be NP-hard; the proof is omitted due to space constraints.

III. Two-step Approach

The combined problem cannot be mapped directly to problems with well known solutions, although it is similar to some classical problems in the area of resource allocation (e.g. Knapsack and Bin packing [9]). Being NP-hard, the combined problem is complex and it is unsolvable for large size. Indeed, in Table I we report the maximum size of the multi-stage router as the number of back-end routers, for a given number of interfaces per router, for which we were able to obtain a solution in reasonable time (e.g. 5 minutes, the default SNMP statistics collection interval time which can be basis for an estimation of input traffic load).

Since the combined problem is not scalable and cannot be used even for small size multi-stage routers with few links per back-end router, we focus on heuristic solution to improve scalability. In proposing such a solution we split the combined problem in two parts using the *divide and conquer* approach.

Interfaces (per router)	Maximum number of back-end routers	
	combined	two-step
1	716	more than 716
4	30	more than 716
16	12	more than 716

TABLE I
Maximum size of the MILPs in terms of number of routers and interfaces to obtain a solution in reasonable time

We name this solution a *two-step* problem: in the first step, the *router optimization* problem, we focus only on routers optimization, whereas in the second step, the *link optimization* problem, we optimize the number of links on each router selected in the first step.

This approach is much more scalable than the combined problem, since the single steps are smaller in size and easier to solve than their parent problem: as shown in Table I, the two-step problem is two order of magnitude more scalable than the combined problem. In case of combined problem, the maximum size we were able to solve was a configuration with 716 routers each equipped with single interfaces. Indeed, the single steps are still NP-hard, but they are easily mappable to well-known problems. Thus, we can take advantage of existing heuristics and approximation algorithms available in the literature to solve them [9], [10].

We first present the two-step approach and later we compare the quality of the two-step solutions to those of the combined problem to determine the impact of problem splitting on the energy saving. We will not consider approximation algorithms to solve the single steps, but we rely on optimal solutions given by CPLEX solver [11] to evaluate approximations introduced due to problem splitting.

1) Router optimization: The first step is the optimal choice of routers without considering the NICs:

$$\min \quad P^{R\_OPT} = \sum_r P_r \alpha_r \tag{7}$$

$$\text{s.t.} \quad \sum_r t_r = 1 \tag{8}$$

$$t_r T \leq C_r \alpha_r, \quad \forall r \in B \tag{9}$$

$$\alpha_r \in \{0,1\}, t_r \in [0,1] \tag{10}$$

where all variables, constraints and terms have the same meaning as in (1)–(6) except t_{rl} which is redefined as t_r because links are not considered here. This problem is a variation of the well known bin packing problem with splittable item (traffic T) where the bins (routers) have different size (routing capacity C_r). Unlike the classical bin packing problem where the cost of using a bin is the same as the size of the bins, in this scheme the cost (power consumption P_r) of the routers is different from the size (routing capacity C_r). Therefore, the above problem is a bin packing problem with generalized cost and variable sized bins [10] with splittable items.

As previously mentioned, the two-step approach is based on the *divide and conquer* paradigm, thus the information required to globally optimize the system is partitioned among the two steps making them less optimal from a global point of view. To assess the impact of information partitioning, we introduce two different schemes to configure the first step:

- **Router-Power scheme** (NIC^-): no link information is made available to the first step. In this scheme we use (7)–(10) with no modification.
- **Router+Link-Power scheme** (NIC^+): NIC power consumption is considered in the first step.

In the NIC^+ scheme the cost of using a router is defined as its power consumption plus the sum of the power consumption of all of its network cards. Therefore the P_r parameter in (7) is replaced by $P_r^{new} = P_r + \sum_l P_{rl}$ which represents the maximum power consumption of router.

2) Link optimization: The link optimization problem makes use of the solution of the first step (i.e. T_r such that $\sum_r T_r = T$) for each router r to determine the links to be activated *independently on each router* as follows:

$$\min \quad P_r^{L\_OPT} = \sum_l P_{rl} \beta_{rl} \tag{11}$$

$$\text{s.t.} \quad \sum_l t_{rl} = 1 \tag{12}$$

$$t_{rl} T_r \leq C_{rl} \beta_{rl}, \quad \forall l \in L_r \tag{13}$$

$$\beta_{rl} \in \{0,1\}, t_{rl} \in [0,1] \tag{14}$$

where T_r is the portion of the traffic T to be routed by router r, as defined from the router optimization step, and the optimization variable t_{rl} is a portion of T_r sent on link l to router r. As in the first step, this formulation is a generalized cost variable sized bin packing problem, where the links are the bins with cost P_{rl} and size C_{rl} and the traffic is the splittable item T_r.

After solving the second step on each router, the total power consumption of the multi-stage architecture due to the two-step approach is given by $P_{two\_step} = P^{R\_OPT} + \sum_r P_r^{L\_OPT}$.

IV. Results

In this Section we present results obtained using CPLEX to implement the optimization models and we compare combined and two-step solutions against the scenario where no energy saving mechanism is implemented. For the comparison purpose, we consider a multi-stage router architecture with features comparable to that of a Juniper T320 router. The main parameters are as follows:

- LBs and router power consumption $P_{LB} = P_r = 80$ W
- Back-end router routing capacity $C_r = 8000$ Mbps [6];
- Link capacity $C_{rl} = 1000$ Mbps;
- Link power consumption $P_{rl} = 2$ W

As per this profile, the 160 Gbps forwarding capability of Juniper T320 can be realized using 20 back-end routers interconnected through a switch, each back-end router being equipped with eight 1 Gbps single-port NICs. Without energy saving scheme, this network consumes about 3200 W (1920 W by back-end routers) without considering the internal switch and assuming to have 16 LBs each with one 10 Gbps link. However, the energy-saving scheme proposed can save up to 1838 W when a minimal traffic is offered at inputs (i.e. only one back-end router and one of its link are active to guarantee minimal functionalities).

Besides the analysis performed with the above described multi-stage router configuration, we also evaluate independently the impact of the variability of four configuration parameters (i.e. C_r, P_r, C_{rl} and P_{rl}) on the solution quality (measured as the difference between the combined solution and the two-step solution). More precisely, we fix three of the parameters to the above given default values, while we vary the fourth parameter as follows:

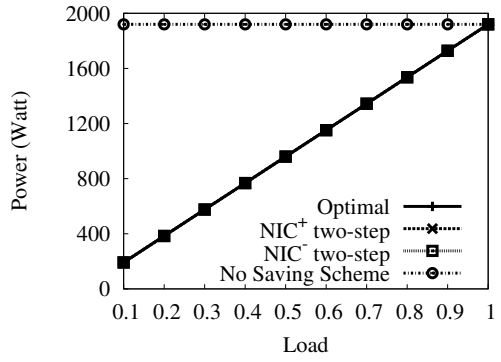


Fig. 2. Load proportional energy saving scheme in back-end routers

- C_r: uniformly distributed (5000 Mbps $\leq$ C_r $\leq$ 10000 Mbps). Ten links (at 1 Gbps) for each router are required to avoid bottleneck.
- P_r: uniformly distributed (60 W $\leq P_r \leq$ 120 W).
- C_{rl}: standard link rates (100 Mpbs with probability $p = 0.25$, 1 Gbps with probability $p = 0.75$).
- P_{rl}: randomly chosen from $\{2, 3, 4\}$ W.

For each of the four above described scenarios, we run the combined and the two-step schemes for loads ranging from 10% to 100% of the total routing capacity of the multi-stage architecture. For each of those load values, 20 random instances were generated and results were averaged over the instances. Furthermore, in the case of the two-step scheme, we evaluate NIC$^-$ and NIC$^+$ schemes under the same configurations. The comparison metric is the total power consumption of the back-end stage (minimum, maximum and average over the 20 instances).

The main outcome of our evaluations are as follows:

1) The energy saving scheme makes the energy consumption of the back-end stage of the multi-stage router *proportional to input load* allowing a large energy saving, as expected (see Fig. 2);
2) The combined and the two-step approaches are very similar. In the worst case considered in our scenarios, the maximum power difference between the two-step approach and the optimal solution is less than 10% (see Fig. 3);
3) NIC$^-$ and NIC$^+$ schemes are usually equivalent, but the NIC$^+$ has a huge impact on the P_{rl} scenario because the heuristic is useful to reduce the negative impact of the greedy approach (see Fig. 3(a) and 3(b)) .

Now we analyze more precisely the impact of each parameter on the proposed optimization techniques.

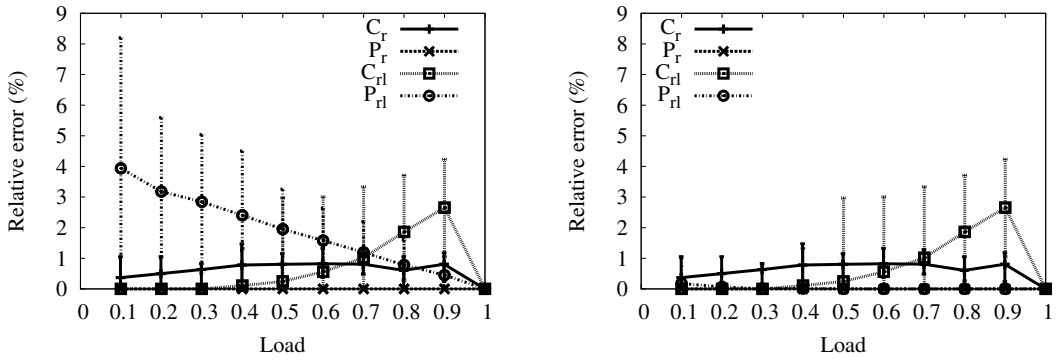
C_r scenario: The effect of C_r variability on the two-step approach is minimal as reported in Fig. 3. The difference is due to the greedy nature of the two-step: In the first step there is no knowledge of the available links, so it happens that the amount of traffic sent to routers by the first step cannot be managed efficiently by the second optimization step on each of the routers. For instance, consider a simple scenario involving two back-end routers R1 ($C_r = 9100$ Mbps, $P_r = 80$ W, two links at 1 Gbps with $P_{rl} = 4$ W) and R2 ($C_r = 7300$ Mbps, $P_r = 80$ W, two links at 1 Gbps with $P_{rl} = 4$ W) only. Observe that the routers are equivalent from the point of view of the objective function of the first step (i.e. P_r). Let us assume an input load $T = 11$ Gbps. A possible optimal solution is to route 9 Gbps to R1 and 2 Gbps to R2 using nine links on R1 and two links on R2 at full capacity and consuming 204 W. However, in the two-step approach since there is no knowledge of links in the first step, the first solution usually maximizes the usage of one of the routers. For instance, one possible choice is to forward 9100 Mbps to R1 and 1900 Mbps to R2 using ten links on R1 and two links on R2 consuming 208 W, 4 W worse than the optimal power consumption.

Thus, the solution of the first step is not always well-suited to efficiently load the links, leading to higher energy consumption. However, the amount of additional energy required is generally small because only few additional links are involved Finally, there is no difference among NIC$^-$ and NIC$^+$, because the routers support the same set of links. In the case of NIC$^+$ the same amount of power is added to all the routers, thus the difference among routers remain the same.

P_r scenario: There is no difference among the two-step and the combined approaches, because the variability is introduced in the optimization parameter included in the objective function. And there are no issues related to the inefficiency in link utilization in the second step as in the previous case, because the total available capacity C_r (8 Gbps) is exactly the sum of the available links (eight links at 1 Gbps). This means that the same routers will be chosen by both schemes, showing no differences in Fig. 3. Furthermore, NIC$^-$ and NIC$^+$ are equivalent for the same reasons explained in previous scenario.

C_{rl} scenario: This scenario highlights the weaknesses of the two-step approach as reported in Fig. 3 especially at high loads. Since all the routers have the same power consumption, it is important to efficiently use the links by sending the right amount of traffic to all the routers as in the C_r scenario. However, this cannot be done by the two-step scheme where all the routers are equivalent in the router optimization step. Furthermore, the difference among optimal and two-step schemes increases with the load, because more and more routers receive a wrong portion of the load. Finally, the NIC$^+$ heuristic cannot improve the solution because all the links have the same power consumption.

P_{rl} scenario: As in the previous scenario, the variability in links highlights the weaknesses of two-step scheme. In this case, all the routers are equally likely to be included in the solution by the two-step scheme. The relative error reported in Fig. 3(a) is decreasing with load because at small loads it is less likely to activate the best routers (which are *randomly* chosen because they are equivalent). However, at high loads, most of the routers are activated; thus, it is less likely to exclude the best routers from the solution. The NIC$^+$ heuristic is very effective because the aggregation of power consumption of links and routers permits to choose more efficiently the best routers giving more priority to routers



(a) NIC- (b) NIC+

Fig. 3. Relative error due to variability in router and link parameters

hosting power-efficient links. As presented in Fig. 3(b), there are still some small differences at low load, but this is due to the fact that the router optimization step is choosing on the aggregate power and not on the single link power. As a consequence, some errors are more likely when few links will be used, as in low load case.

V. Conclusions and future work

We present an energy saving scheme to adapt the energy consumption of multi-stage software router to a traffic load by properly choosing the set of active back-end routers to match incoming load. We defined an optimal problem and we proposed a *two-step* approach to improve the scalability of the solution to make it practical. The simulation results obtained on a realistic scenario show that the solution quality of the proposed scheme is within 10% of the optimal solution in the worst case considered.

Though the energy saving scheme is defined for multi-stage software router, it could easily be adapted to other distributed architectures composed of different parts (e.g. a router with multiple line cards) where energy efficiency of the parts need to be addressed independently.

In broadening the application of the proposed scheme, in the future we will consider multiple link network card scenario and an on-line differential scheme, i.e. a scheme in which the new back-end router configuration is obtained by varying the current configuration.

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] L. Epstein and A. Levin, "An APTAS for generalized cost variable-sized bin packing," *SIAM J. Comput.*, vol. 38, pp. 411–428, April 2008.

[11] "IBM ILOG CPLEX Optimization Studio." [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/ [Accessed: May, 2011]