On the Energy Efficiency of Centralized and Decentralized Management for Reservation-Based Networks

Anne-Cécile Orgerie*, Laurent Lefèvre* and Isabelle Guérin-Lassous*
*ENS de Lyon - LIP (UMR CNRS, INRIA, ENS, UCBL)
46 allée d'Italie 69364 LYON Cedex 07 - France
Email: annececile.orgerie@ens-lyon.fr, laurent.lefevre@inria.fr, isabelle.guerin-lassous@ens-lyon.fr

***Abstract*—Reducing the energy consumption of wired networks has become a key concern for manufacturers of network equipments and network providers. In this paper, we propose an energy-efficient data transfer framework that uses advance bandwidth provisioning and on/off algorithms to put unused nodes of wired networks into sleep mode. This framework is termed as High-level Energy-awaRe Model for bandwidth reservation in End-to-end networkS (HERMES). We explore centralized, decentralized and clustered approaches for managing network resources and bandwidth reservations in this framework. By applying these approaches to HERMES, we evaluate via simulation their efficiency in terms of performance and energy consumption.**

I. Introduction

With network infrastructure accounting for a considerable share of the electricity consumed by current datacenters, reducing the energy consumption of wired networks has become a key concern for equipment manufacturers and providers. When designing networks, energy efficiency should be considered alongside important goals, such as scalability, reliability, response time, low overhead, interoperability and easiness of use. Moreover, the design of policies for reducing power consumption should be adapted to the network characteristics, including its topology, traffic and usage scenario (e.g. peer-to-peer, web servers, VoIP).

This work focuses on improving the energy efficiency of dedicated networks, such as those deployed in data centers, enterprises, across banks [1], and research networks (e.g. UltraScience Net[1]). Unlike the Internet, these networks present more controlled traffic conditions and less intricate topologies. In such networks, most traffic is concentrated on a few links [1], [2] and it consists of relatively large data transfers such as bulk transfers, backup operations and file transfers [3]. By using a reservation system, we can plan and schedule these data transfers in more energy-efficient ways [4].

However, the manner bandwidth reservations are managed can influence the amount of energy consumed by the network infrastructure. Hereafter, we term as management system the system responsible for managing bandwidth reservations in wired networks.

[1] *http://www.csm.ornl.gov/ultranet/*

Different architectural approaches can be applied to building management systems, including centralized, decentralized and clustered (where clusterheads are responsible for the resources of clusters). The clustered scheme is often used for communication in ad-hoc sensor networks [5], and so far it has not been applied to wired networks with the goal of minimizing the energy consumed by a management system. We advocate that this approach should be investigated to determine whether it saves energy when applied to management systems like it does in sensor networks. Therefore, this work compares the three management approaches by applying them to HERMES[4]; a data transfer framework that uses advance bandwidth provisioning and is dedicated to wired networks. We present a simulation-based evaluation of these approaches for different networks and workloads, and show how much energy they consume when managing the reservation system.

Section II presents related work. The architecture of HERMES is detailed in Section III. Then, a description of the different management approaches is given in Section IV. Section V shows the evaluation results obtained with Bookable Network Simulator (BoNeS). Finally, Section VI concludes the paper and presents future work.

II. Related Work

A. Centralized and decentralized management

Work on network management traditionally focuses on either decentralized [6] or centralized approaches [7]. Decentralized approaches are often explored when autonomic management is required (e.g. in P2P networks [6]). Centralized approaches are more suitable for private networks, including campus networks and enterprise networks, where management should support numerous services [7]. In [8], the authors present preliminary results and a brief architecture description of a centralized management system that aims to dynamically optimize the energy consumption by turning off routers while meeting user QoS constraints. Considering wired networks, both approaches present advantages and limitations, but to the best of our knowledge, they have not been compared in terms of energy consumption.

B. Bandwidth Allocation Algorithms

The idea of reserving network resources in advance is not recent [9], but unpredictability of routing behavior has always been an obstacle to its adoption. The emergence of the MPLS (Multi-Protocol Label Switching) standard with traffic engineering and explicit routing features has made it possible to disconnect the reservation management from the network layer, thus leading to an easier inter-operability for the bandwidth reservation systems. Different data transfer scheduling techniques can be used for advance reservation, including: online scheduling where requests are processed as they arrive, and periodic batch scheduling where they are scheduled with certain periodicity [10]. The solutions proposed so far do not consider the network's energy consumption as a major issue that should influence the design of algorithms for network management, reservation scheduling and routing.

C. Green Wired Networking

Despite the ever-increasing power consumption of wired networks [11], their energy demand can be greatly reduced using certain techniques. Studies have shown that network links, especially edge links, are lightly utilized [11], [12]. This fact has led to approaches that take advantage of link under-utilization in order to save energy. The first approach, known as *shutdown*, consists in switching off network equipments (or putting them in sleep mode) when they are not in use [13]. This technique raises several problems: connectivity loss, long to re-synchronization time, and the fact that constantly switching equipments on and off can be more energy consuming than keeping them always on. New mechanisms have been designed to address these issues, including proxy techniques to keep connectivity [14] and re-synchronize both ends of a link [15].

NICs and switches consume less energy when they operate at lower data rates [16]. This observation has resulted in techniques to adjust a link's data rate dynamically according to the load [17]; techniques commonly termed as *slowdown*. We have developed a coordinated model that uses shutdown and slowdown techniques for managing networks, providing end-to-end bandwidth reservations in an energy-efficient way.

III. HERMES ARCHITECTURE

HERMES is a framework for managing bandwidth reservations in dedicated networks. Each data transfer (or flow) between two nodes of the managed infrastructure requires a reservation and should first be submitted to the reservation system via a *gateway* (i.e. the first node connected to the end-user). Then, HERMES schedules the request, informs sender (source) and receiver (destination) about the transfer schedule, and guarantees that the transfer occurs without congestion.

Each network equipment (i.e. router, switch, bridge or NIC) has two agendas per port: for both traffic ways (in and out). An *agenda* stores all future reservations concerning its one-way link. This information is also called the book-ahead interval [18]. A *bandwidth portion* is always kept free on each link for management messages and ACKs. This portion can be either a fixed amount of bandwidth or a fraction of the link's capacity. Each port's agenda contains its reservation status using a *time-bandwidth list* (TB list) which is formed by $(t[i], b[i])$ tuples, where $t[i]$ is the time and $b[i]$ is the bandwidth available. Each $t[i]$ is called an *event* in the agenda. These agendas are not necessarily stored on their corresponding equipments as we will see in the next section.

Each equipment has also an *energy profile* that is determined preliminarily by measuring its energy consumption under different modes: sleep, idle, working at full speed and working at 10% of its capacity. This information can also be obtained from technical specification sheets. The reservation process in HERMES is as follows:

1) a user submits a reservation request (specifying at least data volume and required deadline) to the network-management system;
2) the advance-reservation environment starts the negotiation phase including admission control, reservation scheduling and optimization policies;
3) a notification is sent to the user when his/her request is accepted or rejected, and when it is scheduled;
4) the reservation starts at the scheduled start time and ends at the scheduled end time, which occurs before the user-submitted deadline.

To schedule a reservation, the reservation system needs to gather all agendas and energy profiles in the required path. Then, it merges the agendas to obtain an availability agenda per end-to-end path. Two paths are considered for each reservation to avoid fully busy links. The speed of the merging operation depends linearly on the number of events in the considered agendas for each end-to-end path. This is fast since the agendas are truncated to get only the part between the submission time and the deadline.

The end-to-end availability agenda is scanned using HERMES scheduling algorithm to find the least energy-consuming solution. At each atempt, the solution tries to use as much bandwidth as it can to reduce the reservation's duration, and thus its cost. We estimate the energy consumption of each solution (i.e. event in the agenda to aggregate reservations), we compare each solution, and pick the least energy consuming.

We consider the two shortest paths for each source-destination pair to increase network usage. The scheduling algorithm is launched on the two path agendas, and the least consuming solution is selected. All paths cannot be explored due to computation and communication overheads. Considering two paths is a good trade-off between computation and communication overheads and the increase of network usage.

At the end of a data transfer, if a port remains idle for an interval (over a certain threshold), the port is switched off. If all the ports of a router are switched off and the router will remain idle for a certain period, then the router itself is switched off. To avoid unnecessary on/off cycles, prediction algorithms are used to forecast the next time a link will be used. The prediction algorithms rely on recent history (past agenda) of the port. They are based on average values of past inactivity period durations and feedbacks which are average values of differences between past predictions and the corresponding events in the agenda. These operations (shutdown and prediction) are done in an autonomic way.

In addition to the shutdown techniques, HERMES uses slowdown techniques during the transfers, and dynamically adjusts the transmission rate of each port to the used bandwidth. The reservation process works only if the necessary ports and routers are switched on when the agenda collection is done. However, when they are not used, the network equipments (individual ports or entire routers) are put into sleep mode hence making certain nodes inaccessible. Disruption-Tolerant Networking (DTN) [19] techniques are used to address this issue. DTN is perfectly suitable for this type of scenario where parts of the network are not always available without any guarantee of end-to-end connectivity at any time. The idea is to add a Time-To-Live (TTL) to each end-user request. If the request answer has not returned before the TTL expires, then all the sleeping nodes of the path are awakened and the agenda collection is performed. While the TTL is not expired, the agenda-collection message moves forward along the path until it meets a sleeping node. Then, as long as the TTL has not expired, the message waits in the previous node for the sleeping node to wake up. The message is sent to the sleeping node when it wakes up (wake-up detection managed by the DTN protocol) and continues its way. Hence, hop by hop, the agenda-collection message moves towards its destination.

A more detailed description of HERMES is presented in [4]. This work provides simulation results considering only decentralized management. The results show that on a 500-node network with a 31% workload on the links, HERMES could save 51% of the energy used in the current case (no energy management). Here, we explore other management approaches.

IV. Network management approaches

Gathering the required agendas for request scheduling completely depends on where the agendas are stored. In the decentralized approach, each equipment gets its own agendas (per port and the global agenda). When a request is submitted, it should go on the path from the sender to the receiver and collect the required agendas on the way. In fact, the mechanism collects the agendas as follows. The sender gateway sends a specific management message. The first node to receive it, adds its own availability agenda to the message and forwards it to the two next nodes that are the nearest to the destination. If the network topology is, for example, a simple tree with no redundant link, only one path is available and therefore, the message is sent only to the next node. The agendas of the transmitting ports are also included in the message. Each node includes the required agenda to the message and passes it to the next nodes. At the end, the destination gateway rebuilds the end-to-end paths. In the centralized approach, an equipment stores all agendas and energy profiles. When a request is submitted, a control message is sent to the central node, it proceeds to agenda merging and scheduling and then sends the response to the sender.

The organization in clusters is similar to a hierarchical structure: selected nodes are responsible for other neighboring nodes. The selected nodes are called *cluster-heads*. We call *k-hop clustering* the network organization into clusters where the cluster-head is at maximum at k-hops from the nodes for which it is responsible. The control message in this approach should go to all the cluster-heads responsible for the nodes of the path between the source and the destination. The cluster-head election is done when the network is first initialized, following the algorithm based on node degrees described in [5]. It should be performed each time the network changes (node additions or removals). This requirement is not constraining since dedicated networks do not change frequently. Considering the NSF network as an example, Figure 1 presents this election for all clustered cases: decentralized (0-hop), 1-hop clustering, 2-hop clustering and centralized (3-hop).

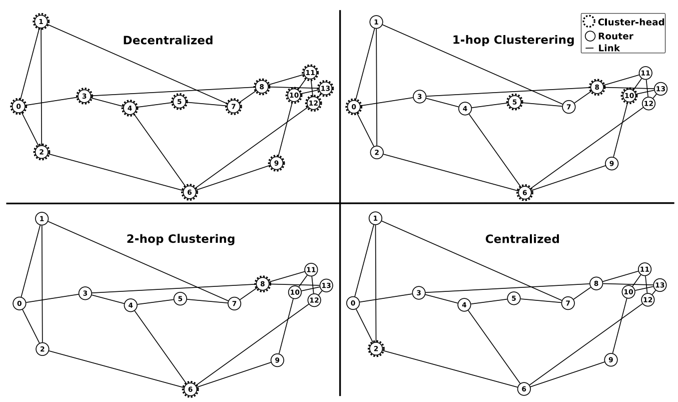

Fig. 1. Examples of cluster-head election on the NSF network.

Similarly to the centralized scheme, in the clustering approach the path of the request is not the same as the path of the reservation. In fact, the reservation path goes from the sender to the receiver whereas the request path goes from the sender to the nodes storing the agendas of the nodes along the reservation path (i.e. the clusterheads).

For example, using the decentralized approach in the network presented in Figure 1, to go from node 4 to node 13, the request uses nodes 4, 6, 12 and 13. In the 1-hop clustering, the request uses nodes 4, 5 (clusterhead of 4), 4, 6 (clusterhead of 6 and 12), 9 and 10 (clusterhead of 13). Hence, the control message ends with all the required agendas. The request path is longer than in the decentralized path. In the 2-hop clustering case, the request uses nodes 4, 6 (clusterhead of nodes 4, 6 and 12), 12, 11 and 8 (clusterhead of node 13). In the centralized approach, the request uses nodes 4, 6 and 2 (clusterhead of all the nodes). In the last two cases (2-hop and centralized), the alternate path that uses nodes 4, 3, 8 and 13, has the same request path as the first path (nodes 4, 6, 12, 13). Therefore, the control messages are aggregated on the same links and no other links need to be woken up. This is not the case for the 1-hop and the decentralized approaches, which require to wake up two distinct paths to get the agendas of the two possible reservation paths.

Because of the on/off capabilities of the network equipments considered in HERMES, these different approaches lead to different overall energy consumptions. While the decentralized approach is more fault-tolerant and scalable, the centralized approach seems more energy-efficient (in terms of sent control messages) and simple to manage. Coordination is necessary to obtain system-wide energy-savings. Hence, the clustering approach could be a good trade-off between advantages and disadvantages of both approaches. In fact, in this approach several requests from different sources can be aggregated along the same path, and thus save energy.

V. Evaluation

To evaluate the different approaches, we have designed in Python a new network simulator (i.e. BoNeS) able to manage bandwidth reservations with these different network management schemes[2]. The simulator takes as input a network-description file (topology, router and link capacities) and network-traffic characteristics (e.g., statistical distribution of inter-arrival submissions, distribution of the reservation durations, source and destination nodes, distribution of the deadlines and TTLs). It generates the network and traffic according to the characteristics given as input. It then simulates, with this traffic and topology, the scheduling algorithm with the different network management approaches and compares them in terms of performance and energy consumption. The generated network traffic consists of requests with:

- submission times distributed according to a log-normal distribution;
- data volumes generated with a negative exponential distribution;
- sources and destinations chosen randomly (equiprobability);
- intervals between submission times and deadlines generated following a Poisson distribution;
- TTLs generated following a Poisson distribution.

The parameters for the probability distributions of the different traffic characteristics can be configured. The distributions presented here have been based on the results presented in [20].

The energy consumption of a network equipment (i.e. switch or router) depends on: the type of equipment, the number of ports, the port transmission rates (with ALR), and the employed cabling solutions [21]. Thus, for each router, we modeled the energy consumption with two values for the chassis power ($P_{chassis}$) depending on whether it is on or off (150 W and 10 W respectively for a 1Gbps router). We use several values for the port power consumption P_{port}: one for when it is off, one for when it is idle (working at the lowest transmission rate), and one for each possible transmission rate. Time to boot and to shut down ports and routers are considered.

[2]Some experiments of this article were performed on the Grid'5000 platform (http://www.grid5000.fr).

For the evaluation, we propose three networks randomly generated using the Molloy and Reed method [22]. The first network called *Net1* has 100 nodes and 259 links, and its diameter is 5. The second network called *Net2* has 100 nodes and 1029 links, and its diameter is 3. The third network called *Net3* has 200 nodes and 524 links, and its diameter is 6.

TABLE I
ENERGY CONSUMPTION FOR 10% WORKLOAD ON *Net1*.

Approach	centr.	1-hop	2-hop	3-hop	decentr.
Average (Wh)	13 970	14 091	13 948	13 794	13 897
Standard deviation	70	120	81	45	49
Messages	47 837	114 880	116 521	36 591	44 574

Each experiment, representing one hour of simulated time, has been launched 40 times on 40 different traffic files generated with the same characteristics to obtain representative average values. For each experiment, we evaluate the energy consumption of the whole network for the duration of the experiment taking into account the on/off and working consumptions of each equipment, and the number of 1-hop messages needed by all the reservations (i.e. if a message needs to traverse three links, it counts as three messages).

TABLE II
ENERGY CONSUMPTION FOR 10% WORKLOAD ON *Net2*.

Approach	centr.	1-hop	decentr.
Average (Wh)	15 467	15 578	15 565
Standard deviation	11	16	14
Messages	145 465	258 814	130 924

Table I presents the results for *Net1* with a 10% workload on each node. For this network with this traffic, there is a difference of almost 400 Wh between 1-hop and 3-hop (about 3% of the total consumption) for a 1-hour experiment. The 3-hop clustering presents the best results in terms of energy consumption and number of messages, beating the centralized and the decentralized approaches. As for the NSF Network example, the worst case in terms of number of messages is the 1-hop approach.

TABLE III
ENERGY CONSUMPTION FOR 10% WORKLOAD ON *Net3*.

Approach	centr.	1-hop	2-hop	3-hop	4-hop	decentr.
Average (Wh)	28 470	28 458	28 228	28 041	28 103	28 081
Deviation	80	148	125	98	63	65
Messages	107 999	265 648	310 038	290 247	90 401	87410

Table II presents the results for *Net2* for a 10% workload. In this case, with a highly connected network (node degree of 20 on average), the best solution is the decentralized approach. With this traffic, it can save 100 Wh per hour compared to the centralized approach (2-hop) and 110 Wh compared to the 1-hop clustering approach. The number of messages in the 1-hop case is almost twice the number in the 2-hop case. Yet, their energy consumptions are really close. As the network is really concentrated and the traffic is not intensive, most links are not necessary. In the decentralized approach, less links are used since the reservation path and the request path are the same. Thus, control messages can be aggregated with reservations, whereas in the decentralized approach (2-hop), the request path is shorter on average but different from the reservation path. Thus, more links are used and the traffic is less aggregated.

TABLE IV
ENERGY CONSUMPTION FOR 40% WORKLOAD ON *Net1*.

Approach	centr.	1-hop	2-hop	3-hop	decentr.
Average (Wh)	15 213	14 975	15 498	15 839	15 813
Deviation	119	147	104	37	30
Messages	173 070	419 753	418 856	131 721	160 333

Table III presents results similar to Table I: the 3-hop approach consumes the least power followed by the centralized approach, and decentralized approaches are the most energy consuming. Thus, increasing the network size does not affect the ranking of the approaches for a low usage (10%).

TABLE V
ENERGY CONSUMPTION FOR 75% WORKLOAD ON *Net1*.

Approach	centr.	1-hop	2-hop	3-hop	decentr.
Average (Wh)	16 578	16 003	16 344	16 655	16 607
Deviation	55	121	75	17	30
Messages	336 615	804 771	808 912	256 100	311 872

Table IV presents the results for *Net1* for a 40% workload. The most energy efficient approach here is by far the 1-hop contrary to what we observed with a low workload (Table I). When the workload increases further (see Table V), the ranking is almost the same and the best solution is still the 1-hop approach, whereas the worst solution is the 3-hop approach. With medium and high workloads, the control messages do not require nodes to wake up as most of the network is already powered on due to the high traffic. These results show that clustering approaches should be considered alongside classical approaches to save energy.

VI. Conclusion and Future Work

HERMES allows impressive energy consumption reductions in reservation-based networks [4]. However, the network management system has a non-negligible influence on the overall energy consumed by the network. We have identified network density and workload as factors that have an important impact on the energy consumption in reservation-based networks. These two network characteristics should be used to determine the management system to apply to the targeted network in order for it to be energy-efficient. Decentralized management performs better on concentrated networks with low traffic. Whereas big clustering (i.e. with a large cluster size) approaches are more energy-efficient on lightly-used networks which are not concentrated. However, as the network traffic increases, small clustering techniques become more energy-efficient.

We are currently studying mechanisms to adapt the network management system configuration to the workload in an autonomic way. Generally, the density of wired networks does not vary a lot, while the workload can present daily or weekly patterns with peak traffic and slack periods.

Acknowledgment

We thank Marcos Dias de Assunção (INRIA, France) for his feedback and useful comments on this work.

References

[1] K. Soramki, M. Bech, J. Arnold, R. Glass, and W. Beyeler, "The Topology of Interbank Payment Flows," *Physica A: Statistical Mechanics and Its Applications*, vol. 379, no. 1, pp. 317–333, 2007.

[2] T. Benson, A. Akella, and D. Maltz, "Network traffic characteristics of data centers in the wild," in *Conference on Internet measurement (IMC)*, 2010, pp. 267–280.

[3] R. Pang, M. Allman, M. Bennett, J. Lee, V. Paxson, and B. Tierney, "A first look at modern enterprise traffic," in *Conference on Internet Measurement (IMC)*, 2005.

[4] A.-C. Orgerie, L. Lefèvre, and I. Guérin-Lassous, "Energy-Efficient Bandwidth Reservation for Bulk Data Transfers in Dedicated Wired Networks," *Journal of Supercomputing*, 2011.

[5] S. Basagni, "Distributed clustering for ad hoc networks," in *International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN '99)*, 1999, pp. 310–315.

[6] J. Nobre and L. Granville, "Consistency maintenance of policy states in decentralized autonomic network management," in *Network Operations and Management Symposium (NOMS)*, 2010, pp. 519–526.

[7] X. Wang, L. Wang, B. Yu, and G. Dong, "Studies on Network Management System framework of Campus Network," in *International Asia Conference on Informatics in Control, Automation and Robotics (CAR)*, 2010, pp. 285–289.

[8] E. Gelenbe and S. Silvestri, "Optimisation of Power Consumption in Wired Packet Networks," in *Quality of Service in Heterogeneous Networks*, ser. LNCS. Springer, 2009, vol. 22, pp. 717–729.

[9] W. Reinhardt, "Advance Reservation of Network Resources for Multimedia Applications," in *International Workshop on Multimedia (IWAKA 1994)*, 1994, pp. 23–33.

[10] Y. Li, S. Ranka, and S. Sahni, "In-advance path reservation for file transfers In e-Science applications," in *IEEE Symposium on Computers and Communications (ISCC 2009)*, 2009, pp. 176–181.

[11] K. Christensen, C. Gunaratne, B. Nordman, and A. George, "The next frontier for communications networks: power management," *Computer Communications*, vol. 27, no. 18, pp. 1758–1770, 2004.

[12] A. Odlyzko, "Data Networks are Lightly Utilized, and will Stay that Way," *Review of Network Economics*, vol. 2, 2003.

[13] M. Gupta and S. Singh, "Greening Of The Internet," in *SIGCOMM '03*, 2003, pp. 19–26.

[14] B. Nordman and K. Christensen, "Proxying: The Next Step In Reducing IT Energy Use," *Computer*, vol. 43, no. 1, pp. 91–93, 2010.

[15] M. Gupta and S. Singh, "Dynamic Ethernet Link Shutdown For Energy Conservation On Ethernet Links," in *International Conference On Communications (ICC '07)*, 2007, pp. 6156–6161.

[16] C. Gunaratne, K. Christensen, and B. Nordman, "Managing Energy Consumption Costs In Desktop PCs And LAN Switches With Proxying, Split TCP Connections, And Scaling Of Link Speed," *International Journal Of Network Management*, vol. 15, no. 5, pp. 297–310, 2005.

[17] M. Bennett, K. Christensen, and B. Nordman, "Improving The Energy Efficiency Of Ethernet: Adaptative Link Rate Proposal," 2006.

[18] L.-O. Burchard, "Networks with Advance Reservations: Applications, Architecture, and Performance," *Journal of Network and Systems Management*, vol. 13, no. 4, pp. 429–449, 2005.

[19] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, and P. McDonald, "When TCP Breaks: Delay- and Disruption- Tolerant Networking," *IEEE Internet Computing*, vol. 10, no. 4, pp. 72–78, 2006.

[20] D. Ersoz, M. Yousif, and C. Das, "Characterizing Network Traffic in a Cluster-based, Multi-tier Data Center," in *International Conference on Distributed Computing Systems (ICDCS)*, 2007.

[21] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A Power Benchmarking Framework For Network Devices," *Networking*, vol. 5550, pp. 795–808, 2009.

[22] M. Molloy and B. Reed, "A critical point for random graphs with a given degree sequence," *Random Structures and Algorithms*, vol. 6, pp. 161–179, 1995.