VOL. 10, NO. 7, APRIL 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

PACKET PROCESSING ENGINE WITH FIREFLY SCHEDULING IN GREEN NETWORKING

S. S. Saranya and S. Srinivasan
Computer Science and Engineering, Regional Centre of Anna University, Madurai, India
E-Mail: vasukirajan@gmail.com

ABSTRACT

With the aim of controlling power consumption in core networks, we consider energy-aware devices able to reduce their energy requirements by adapting their performance. We propose new algorithm for scheduling the task to different pipelines to balance the energy consumption in networking. An also we use MILP(Mixed Integer Linear Programming) which is a optimization problem of task assignment, data loading and data movement. The firefly algorithm (FA) is a meta heuristic algorithm, inspired by the flashing behavior of fireflies. The primary purpose for a firefly's flash is to act as a signal system to attract other fireflies. mixed integer linear programming framework that solves the virtual topology problem under the communication delay constraint. An arbitrary optical network has been considered with different distances between the nodes and different link capacities. We are using following steps to minimize the energy consumption ,Packet Segmentation for avoiding the collision in single pipeline, Firefly Algorithm for optimizing the identifying the pipe line , Mixed Integer Linear Programming is for joint optimization of task assignment , data loading and data routing. The purpose of our work is to minimize the energy consumption in overall network.

Keywords: packet segmentation, firefly algorithm, mixed integer linear programming (MILP) framework.

1. INTRODUCTION

The study of power-saving network devices has been based in recent years on the possibility of adapting network energy requirements to the actual traffic load. Indeed, it is well known that network links and devices are generally provisioned for busy or rush-hour load, which typically exceeds their average utilization by a wide margin. Although this margin is seldom reached, network devices are designed on its basis and, consequently, their power consumption remains more or less constant even in the presence of fluctuating traffic load. Thus, the key of any advanced power saving criteria [1] resides in dynamically adapting resources, provided at the network, link, or equipment level, to current traffic requirements and loads. In this respect, current green networking approaches[19] have been based on numerous energy-related criteria, to be applied in particular to network equipment and component interfaces.

Green networking [9] is the practice of selecting energy-efficient networking technologies and products, and minimizing resource use whenever possible. Green networking is a broad term referring to processes used to optimize networking or make it more efficient. This term extends to and covers processes that reduce energy consumption, as well as processes for conserving bandwidth or any other process that will ultimately reduce energy use and, indirectly, cost. The issue of green networking has many important applications, especially as energy becomes more expensive and people become more conscious of the negative effects of energy consumption on the environment. Some of the main strategies associated with green networking involve consolidating devices or otherwise optimizing a hardware setup.

Software virtualization [10] and efficient server use can contribute to this general goal. Green networking could also include such diverse ideas as remote work location, energy use in buildings housing hardware, or other peripheral aspects of a network infrastructure. Ideas associated with green networking also address tech services or user relationships that may ultimately be built on a network. This includes green search or studies of the energy use of search engines, along with many other kinds of analysis of modern networks and systems. According to a number of studies, IT can consume up to 2 percent of a nation's total energy production. Much of the scientific data carried by ESnet and fellow research and education (R&E) networks is either generated by simulations at supercomputing centers, or ends up at data centers for storage, analysis, and access by others. These have been the subject of intense energy efficiency efforts, but the networks that connect them have been largely overlooked. Moreover, in computing-intensive scientific fields ranging from high-energy physics to climate studies, the amount of data that traverses scientific networks continues to rise exponentially. In routing, the forwarding engine [3], sometimes called the data plane, defines the part of the router architecture that decides what to do with packets arriving on an inbound interface.

Transmit data as fast as possible, return to Low-Power Idle– Highest rate provides the most energy-efficient transmission (Joules/bit)– LP_IDLE consumes minimal power (Watts).Energy savings come from cycling between Active & Low-Power Idle – Power is reduced by turning OFF unused circuits during LP_IDLE (e.g. portions of PHY, MAC, interconnects, memory, CPU).Energy consumption scales with bandwidth utilization. Raffaele Bolla *et al*. [14] raise the same concern in their work save energy by scaling their traffic processing capacities through AR and LPI mechanisms.

The rest of the paper is organized as follows: Section II describes the related work of less energy consumption Based on Green network technique. Section

III portrays the Analysis of proposed methods. The Experimental results are shown in the Section IV.

2. RELATED WORKS

FLARE method [14] is possible to systematically slice a TCP flow across multiple paths without causing packet reordering. Srikanth Kandula et al. (2007) FLARE, a new traffic splitting algorithm. FLARE exploits a simple observation. Consider load balancing traffic over a set of parallel paths. If the time between two successive packets is larger than the maximum delay difference between the parallel paths, one can route the second packet and subsequent packets from this flow on any available path with no threat of reordering. Thus, instead of switching packets or flows, FLARE switches packet bursts, called owlets. Dynamic load balancing needs schemes that split traffic across multiple paths at a fine granularity. Current traffic splitting schemes, however, exhibit a tussle between the granularity at which they partition the traffic and their ability to avoid packet reordering. Packet-based splitting quickly assigns the desired load share to each path

Power management capabilities [13] inside architectures and components of network equipment. R. Bolla *et al*.(2007) considering the two main kinds of power management hardware support, today available in the largest part of COTS processors and under rapid development in other hardware technologies [15] (e.g., network processors, ASIC and FPGA). These power management technologies respectively allow minimizing power consumption when no activities are performed (namely, "idle" optimizations), and to modify the trade-off between performance and energy when the hardware is active and performing operations (namely, "power state" optimizations). These kinds of power management support are generally realized at the hardware layer by powering off sub-components, or by changing the silicon operating frequency and voltage.

Dynamic Traffic method [2] proposes an approach that reconfigures the network in order to reduce the energy consumption, based on the current traffic load. This paper main idea is to switch on a minimum amount of necessary switches/routers and links to carry the traffic. Adam Markiewicz et al. first formulate the problem as a mixed integer linear programming (MILP) problem [6] and further present a heuristic method, so called Strategic Greedy Heuristic, with four different strategies, to solve the problem for large networks.

Load Migration method [11] With wireless resource virtualization, multiple Mobile Virtual Network Operators (MVNOs) can be supported over a shared physical wireless network and traffic loads in a Base Station. Xiang Sheng et al. a general optimization framework to guide algorithm design, which solves two sub problems, channel assignment and load allocation, in sequence. For channel assignment, this paper present a approximation algorithm For load allocation, we present a polynomial-time optimal algorithm for a special case where BSs are power-proportional as well as two effective heuristic algorithms for the general case. In addition, this paper presents an effective heuristic algorithm that jointly solves the two sub problems.

Fire resource scheduling model [16] on the ground of major hazards, where time limitation of major hazards and actual situation of fire resource can be taken into account on all sides. Thus, in line with the bear able loss and time limitation of major hazards, GOU Gang et al. choose fire stations which own certain fire cover ability and relatively low cost for distance as target combination. Fire stations arrive at accident points and conduct rescue work, so as to minimize the loss in whole accident.

Rate adaptation method according to recent measurement studies, we consider a discrete step increasing function for link power consumption. Jian Tang et al. address both the single and multiple communication session cases and formulate them as two optimization problems, namely, the Single-session Flow allocation with Rate Adaptation Problem and the Multisession Flow Allocation with Rate Adaptation Problem first show that both problems are NP-hard and present a Mixed Integer Linear Programming (MILP) formulation [11] for the MF-RAP to provide optimal solutions. Then we present a2-approximation algorithm for the SF-RAP, and a general flow allocation framework as well as an LP-based heuristic algorithm for the MF-RAP.

Linux kernel network subsystem [4] the Tx/Rx Soft IRQ and Q disc are the connectors between the network stack and the net devices. A design limitation is that they assume there is only a single entry point for each Tx and Rx in the underlying hardware. Although they work well today, they won't in the future. Modern network devices (for example, E1000 and IPW 2200 equip two or more hardware Tx queues to enable transmission parallelization or MAC-level QoS. These hardware features cannot be supported easily with the current network subsystem. Z. Yi et al. (2007) describes the design and implementation for the network multi queue patches submitted to mailing lists early this year, which involved the changes for the network scheduler, Q disc, and generic network core APIs.

3. ANALYSIS OF PROPOSED METHODS

A pipeline is a set of data processing elements connected in series, where the output of one element is the input of the next one.

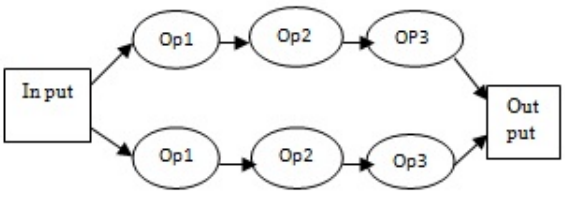

Figure-1. Parallel pipeline.

Figure-1.shows the elements of a pipeline are often executed in parallel or in time-sliced fashion; in that case, some amount of buffer storage is often inserted between elements. The packet processing system is specifically designed for dealing with the network traffic.

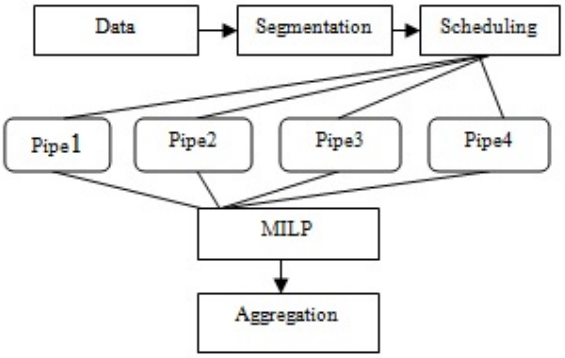

Figure-2. System architecture.

Figure-2 shows System Architecture represents Parallel Processing of different pipe lines. In this system, Fire fly Scheduling algorithm for effectively schedule the input traffic load for load balancing. The Distributed Load processed by the different pipelines. MILP(Mixed Integer Linear Programming) Joint Optimize the Task Assignment and Data Loading and Data Movement.

Packet segmentation improves network performance by splitting the packets in received Ethernet frames into separate buffers. Packet segmentation may be responsible for splitting one into multiple so that reliable transmission of each one can be performed individually. Segmentation may be required when the data packet is larger than the maximum transmission unit supported by the network.

The packet processing system can be equipped in any layer of the network, either in the high end core routers or in the LAN switches. The flexibility of the system comes from the programmable elements within it, i.e. NPs. And a series of stacked network protocols guarantee its capability to achieve the performance specification.

Fire fly algorithm is used for packet scheduling. The firefly algorithm [18] is a meta heuristic algorithm, inspired by the flashing behaviour of fireflies. The primary purpose for a firefly's flash is to act as a signal system to attract other fireflies. In task assignment process, packets distribute across parallel pipe lines. In this Module, segmented Data chunks assigned into Queue for processing of data. This manages Work load distribution to multiple parallel pipelines. This module words at transmitting end.

a) Algorithm

The firefly algorithm is a meta heuristic algorithm [7], inspired by the flashing behaviour of fireflies. The primary purpose for a firefly's flash is to act as a signal system to attract other fireflies. Xin-She Yang [5] formulated this firefly algorithm by assuming:

All fireflies are unisexual, so that one firefly will be attracted to all other fireflies;

Attractiveness is proportional to their brightness, and for any two fireflies, the less bright one will be attracted by (and thus move to) the brighter one; however, the brightness can decrease as their distance increases;

If there are no fireflies brighter than a given firefly, it will move randomly.The brightness should be associated with the objective function.

Firefly algorithm is a nature-inspired meta heuristic optimization algorithm.

b) Algorithm description

The pseudo code can be summarized as:

Begin

1) Objective function:

$f(x), \quad x = (x_1, x_2, \ldots, x_d)$

2) Generate an initial population of fireflies $x_i \ (i = 1, 2, \ldots, n)$

3) Formulate light intensity so that it is associated with f (for example, for maximization problems, or simply)

4) Define absorption coefficient γ

While (t < Max Generation)

for i = 1 : n (all n fireflies)

for j = 1 : n (n fireflies)

if ,

move firefly i towards j;

end if

Vary attractiveness with distance r via exp ;

Evaluate new solutions and update light intensity;

end for j

end for i

Rank fireflies and find the current best;

end while

Post-processing the results and visualization;

End

The main update formula for any pair of two fireflies x_i x_j and is

$$x_i^{t+1} = x_i^t + \beta_0 \exp[-\gamma r_{ij}^2](x_j^t - x_i^t) + \alpha_t \epsilon_t$$

α_t ϵ_t

Where, is a parameter controlling the step size, while is a vector drawn from a Gaussian or other distribution.

It can be shown that the limiting case corresponds to the standard Particle Swarm Optimization (PSO). In fact, if the inner loop (for j) is removed and the brightness is replaced by the current global best, then FA essentially becomes the standard PSO.

The should be related to the scales of design variables. Ideally, the term should be order one, which requires that should be linked with scales. For example, one possible choice is to use where is the average scale of the problem. In case of scales vary significantly, can be considered as a vector to suit different scales in different dimensions. Similarly, should also be linked with scales. For example,

The pipe line is a client server processing system. Incoming flows can be handled by any subset of the pipelines. Each client sent the data to server for processing. The processing is held in server and returns the result back to server. The AR and LPI mechanisms for each pipeline to dynamically manage the engine configuration in order to optimally balance its energy consumption with respect to network performance.

New algorithm for Mixed Integer Linear programming [6] for optimizing the task assignment and data loading. Data collected from each pipe line and aggregated together. Data aggregation is any process in which information is gathered and expressed in a summary form. A common aggregation purpose is to get more information about particular groups based on specific variables. This module works at receiving end. The outcome must be the aggregated result. This aggregated result getting from the processing of parallel pipelines. Data aggregations are used to reduce unnecessary data communication and further to increase the duration of wireless sensor networks with energy constrains. Data aggregations are used to reduce the amount of data communication by merging sensor nodes in networks.

c) MILP

Input: (1)No of chunks on server, (2) flow of link (3) weight on link.
Output: Total Cost.
For (each server j)

$$C_{total} = \sum_{j \in J} x_{j,P_j} + \sum_{j \in J} \sum_{k \in K} \sum_{(u,v) \in E} f_{jk}^{(u,v)} \cdot w^{(u,v)}$$

C total is a total cost on each server.

$$x_{jk} = \begin{cases} 1, \text{if chunk } k \text{ is placed on server } j \\ 0, \text{Otherwise} \end{cases} \quad (1)$$

A binary variable indicating if chunk k on server j or not

$$x_j = \begin{cases} 1, \text{if this server is activated} \\ 0, \text{Otherwise.} \end{cases} \quad (2)$$

A Binary variable Xj denoted whether server is activated or not.
Pj is the P chunks on server j

$$f_{jk}^{(u,v)} \leq y_{jk} \cdot r_{jk} \cdot \forall (u,v) \in E, u \quad e] \quad (3)$$

flow of link between servers j and k
W – weight of link from source to destination

$$r_{jk} \leq r^{(u,v)} \cdot z_{jk}^{(u,v)} - 1 - z_{jk}^{(u,v)}, \forall (u,v) \in E, j \in J, k \in K$$

is a transmission rate on link.
End for

4. EXPERIMENTAL RESULTS

This section describes the performance analysis to validate the proposed algorithm. Experimental results demonstrate the efficiency of the proposed Firefly algorithm.

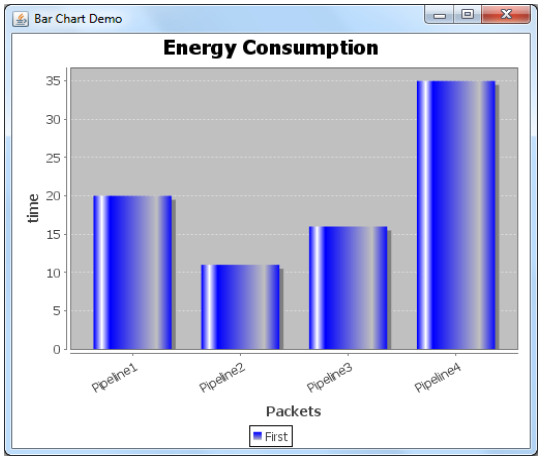

Figure-3. Energy consumption.

Figure-3 depicts the Energy Consumption in parallel pipe line .The Energy consumption varies in parallel pipelines in accordance with time. In this work, Incoming packet are segmented into multiple small packets and allocated to different pipelines. These packets assigned to pipe lines based on size of the chunks by using fire fly algorithm. The data packet 4 take 35 sec for processing and the data packet 2 take 10 sec for processing. The less amount of time represents the low energy consumption. Data packet 2 consumes less energy.

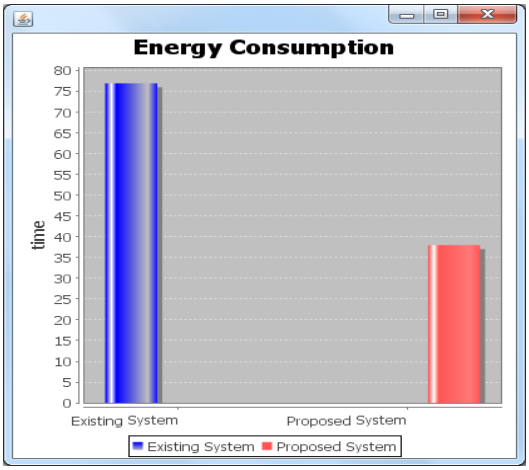

Figure-4. Energy consumption proposed model.

Figure-4 Energy consumption depicts the comparison of existing system and proposed system. In existing system, no scheduling algorithm is considered. So we cannot effectively perform the load balancing. We propose new scheduling algorithm that schedule the packets to different pipe lines based on the capacity of pipeline and chunks. In this graph, we show the energy consumption in existing system, and proposed system.

5. CONCLUSIONS

In this paper, we propose new scheduling algorithm to minimize the energy consumption in Parallel Pipe line System. The firefly algorithm (FA) is a meta heuristic algorithm, inspired by the flashing behavior of fireflies. The primary purpose for a firefly's flash is to act as a signal system to attract other fireflies. Firefly-based algorithms for scheduling task graphs and job shop scheduling requires less computing than all other meta heuristics. Firefly algorithm can solve optimization problems in dynamic environments very efficiently. The achieved results demonstrate how the proposed model can effectively represent energy- and network-aware performance indexes. Moreover, also an optimization procedure based on the model has been proposed and experimentally evaluated. We Propose Mixed Integer Linear Programming which is a joint optimization problem of Task Assignment, Data Movement and Data Loading. These two approaches implemented in our approach for minimizing energy consumption in parallel pipe lines.

REFERENCES

[1] J. Kennedy and R. Eberhart. 1995. Particle swarm optimisation, in: Proc. of the IEEE Int. Conf. on Neural Networks, Piscataway, NJ, pp. 1942-1948.

[2] Y. K. Agarwal. 2002. Design of capacitated multi-commodity networks with multiple facilities, Operations Research, Vol. 50, No. 2, 2002, pp. 333–344.

[3] S. Kandula, D. Katabi, S. Sinha and A. Berger. 2007. "Dynamic load balancing without packet reordering," Comput. Commun. Rev., vol. 37, pp. 51–62, March.

[4] Z. Yi and P. Waskiewicz. 2007. "Enabling Linux network support of hardware multiqueue devices, "in Proc. Linux Symp., Ottawa, ON, Canada, Jun., vol. 2, pp. 305–310.

[5] X. S. Yang. 2008. Nature-Inspired Metaheuristic Algorithms, Luniver Press, UK.

[6] L. Chiaraviglio, M. Mellia and F. Neri. 2009. Reducing power consumption in backbone networks, Proceedings of ICC.

[7] R. Bolla, R. Bruschi, F. Davoli and A. Ranieri. 2009. "Energy-awareperformanceoptimizationfornext-generationgreennetworkequipment,"in Proc. 2nd ACM SIGCOMM PRESTO, Barcelona, Spain, August. pp. 49–54.

[8] X. S. Yang. 2009. Firefly algorithms for multimodal optimisation, Proc. 5th Symposium on Stochastic Algorithms, Foundations and Applications, (Eds. O. Watanabe and T. Zeugmann), Lecture Notes in Computer Science, 5792: 169-178.

[9] "Low Energy Consumption NET works (ECONET) project," 2010 [Online]. Available: http://www.econet-project.eu

[10] B. Heller *et al.* 2010. ElasticTree: saving power in data center networks, Proceedings of USENIX NSDI.

[11] "Energy eFFIcient te Chnolog IEs for the Networks of Tomorrow (EFFICIENT) project," 2010 [Online]. Available: http://www.tnt.dist. unige.it/efficient.

[12] X. S. Yang. 2010. Engineering Optimisation: An Introduction with Metaheuristic Applications, John Wiley and Sons, USA.

[13] A.Bolla and R. Bruschi. 2011. "Energy-aware load balancing for parallel packet processing engines," in Proc. 1st IEEE GREENCOM, September. pp. 105–112.

[14] R. Bolla, R. Bruschi, A. Carrega and F. Davoli. 2011. "Greennetworktechnologies and the art of trading-off," in Proc. 30th IEEE INFOCOM Workshops, Shanghai, China, April. pp. 301–306.

[15] R. Bolla, R. Bruschi, F. Davoli and F. Cucchietti. 2011. "Energy efficiency in the future Internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures," IEEE Commun. Surveys Tut., vol. 13, no. 2, pp. 223–244, 2nd Quart.

[16] S. Palit, S. Sinha, M. Molla, A. Khanra and M. Kule. 2011. A cryptanalytic attack on the knapsack cryptosystem using binary Firefly algorithm, in: 2nd Int. Conference on Computer and Communication Technology (ICCCT), 15-17 September 2011, India, pp. 428432.

[17] "Greeningthe Network (GreenNet) project," 2012 [Online].Available: http://www.tnt.dist.unige.it/greennet.

[18] S. Nandy, P. P. Sarkar and A. Das. 2012. Analysis of nature-inspired firefly algorithm based back-propagation neural network training, Int. J. Computer Applications, Vol. 43, no. 22, p. 816.

[19] Raffaele Bolla, Roberto Bruschi, Alessandro Carrega, and Franco Davoli. 2014. "Green Networking With Packet Processing Engines: Modeling and Optimization" IEEE/ACM Transaction Networking, Vol. 22, No.1, February.