International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 4 | ISSN : 2456-3307

Green Internet with a Hop-by-Hop Routing method

K. Hasna
MCA Sri Padmavathi College Of Computer Sciences And Technology Tiruchanoor, Andhra Pradesh , India

ABSTRACT

We look at the fundamental problem of designing congestion control protocols for background traffic with the minimum impact on short tcp flows while achieving a definite desired average throughput over time. In this paper we study energy conservation within the internet. we have a tendency to observe that totally different traffic volumes on a link may end up in several energy consumption; this is often in the main as a result of such technologies as trunking (IEEE 802.1AX), adaptive link rates, etc. we have a tendency to style a inexperienced net routing theme, wherever the routing will lead traffic in a approach that's green. We have a tendency to disagree from previous studies wherever they switch network elements, like line cards and routers, into sleep mode. We have a tendency to don't prune the web topology. We have a tendency to initial develop a power model, and validate it exploitation real business routers. Rather than developing a centralized optimization rule, which needs extra protocols like MPLS to hap within the net, we elect a hop-by-hop approach. It’s so a lot of easier to integrate our theme into the present net. We have a tendency to increasingly develop 3 algorithms, that are loop-free, substantially reduce energy consumption, and jointly think about green and QoS requirements like path stretch. We have a tendency to more analyze the facility saving ratio, the routing dynamics, and therefore the relationship between hop-by-hop green routing and QoS requirements.

Keywords: hop-by-hop green routing, throughput, protocols.

I. INTRODUCTION

The substantial power consumed by a network has become a critical issue for Internet Service Providers (ISPs). It has been reported that the total energy used by the Information and Communication Technology is responsible for a significant fraction of the world total electricity consumption, ranging between 2% and 10%. With the proliferation of new Internet services, such as social networking and cloud computing, this proportion has increased rapidly in recent years. The tremendous energy consumption of large-scale networks will become a stumbling block to their further developments unless energy efficiencies can be significantly improved.

A great deal of research has been carried out for network energy efficiency based on two mechanisms: speed scaling and power-down. These mechanisms are considered to be basic device-level energy saving approaches and have already been applied in industry. However, without coordinating with other devices, an energy-efficient device in a network can only make local decisions, which may be far from the global optimum, leading to limited network-wide energy savings. Therefore, researchers have started to propose network level energy saving methods based on device-level mechanisms. These network-wide approaches usually assume some given global traffic matrices in a target network and design traffic-shaping strategies to route the traffic flows. However, global traffic matrices are not easy to obtain, even in a small scale network. Moreover, most of the approaches are centralized approaches, whereby global information needs to be gathered, decisions are made by a central controller, and then disseminated to network nodes. This centralized model produces many scalability- and vulnerability-related issues when being applied to production networks. As a result, a decentralized approach is a more scalable and flexible choice to implement and apply to real networks.

When designing a new network protocol or applying a new routing scheme, it is very important for an ISP to consider traffic delays. Unfortunately, many existing proposals for energy savings have failed to take reduced traffic delay as a design goal. As a result, although considerable energy savings can be achieved, the traffic delays may be dramatically increased, which consequently leads to very poor user experience. A good energy-efficient design should be able to achieve an optimal balance between energy efficiency and traffic delay.

Motivation. In this paper, our objective is to obtain increased network-wide energy efficiency while considering traffic delays. A decentralized network routing scheme is developed that simultaneously possesses the following properties:

- ✓ High scalability: With the scale of networks continuously growing, compared with centralized strategies, a decentralized scheme is a more scalable and flexible choice. Specifically, the design needs no centralized controllers, thus avoids a single point of failure and brings high scalability in its implementation.
- ✓ Requires no real-time global traffic matrices: The route computation process in the proposed scheme should not require global traffic matrix from the network. Each node only needs to continuously monitor and react according to the real-time traffic loads of the links attached to its own, which is very easy to implement.
- ✓ Traffic delay: The proposed scheme takes both energy efficiency and traffic delay as design considerations.

II. PROPOSED SYSTEM

GREEN-HR ALGORITHMS:-

We now study hop-by-hop green routing (Green-HR). We first propose a path weight and a baseline algorithm Dijkstra-Green-B to achieve loop-free. We then study some intrinsic relationships between link weights and power consumption, and develop an advanced algorithm DijkstraGreen-Adv that improves energy conservation. We further develop algorithm Dijkstra-Green that concurrently considers energy conservation and path stretch.

Dijkstra-Green-B Algorithm:-

A Dijkstra-oriented algorithm can then be developed to achieve loop-free hop-by-hop routing. A preliminary observation is that though we cannot choose the "greenest" paths, for energy conservation from the whole network point of view, we should not choose a path that is too long either, since it accumulatively consumes more energy.

We thus set the weight as follows. For each link in the path to destination node d, we assign an estimated traffic volume or "virtual traffic volume". We compute the virtual traffic volume by posing an exponential penalty to a start traffic volume for each additional hop.

Algorithm 1. Algorithm Dijkstra-Green-B()

Input: $G(V, E)$, s, d, P, x_0^v, β;
Output: the green path from s to d which is stored in $\varphi[]$;

for each node $v \in V$
  $w[v] \Leftarrow \infty; \varphi[v] \Leftarrow null; h[v] \Leftarrow \infty;$
$Q \Leftarrow V; w[d] \Leftarrow 0; h[d] \Leftarrow 0;$
while $Q \neq \phi$
  $u \Leftarrow \text{Extract\_Min}(Q);$
  if $u = s$
    return $\varphi[]$;
  for each node $v \in N(u)$
    $x \Leftarrow x_0^v \cdot \beta^{h[u]};$
    $\varpi \Leftarrow P_{(v,u)}(x);$
    if $w[u] + \varpi < w[v]$
      $\varphi[v] \Leftarrow u;$
      $w[v] \Leftarrow w[u] + \varpi; h[v] \Leftarrow h[u] + 1;$
    else if $w[u] + \varpi = w[v]$ **and** $h[u] + 1 < h[v]$
      $\varphi[v] \Leftarrow u; h[v] \Leftarrow h[u] + 1;$
return null; // d is unreachable

The computation complexity of Dijkstra-Green-B is the same as that of the standard Dijkstra in the worst case, i.e., $O(|E| + |V|\log|V|)$. However, the algorithm can stop once the path from s to d is finished so the complexity in the best case is Oð1Þ. Note that Dijkstra-Green-B computes the routing for one destination. However, routings of different destinations are independent from each other, so parallel processing can be used to accelerate the computing. We can expect that the running time is less than that of Dijkstra.

Dijkstra-Green-Adv Algorithm:-

Now we develop the Dijkstra-Green-Adv algorithm based on the improvements above. Note that DijkstraGreen-B in Section is a baseline algorithm, which focuses on loop-free hop-by-hop routing while considering energy conservation. Dijkstra-Green-Adv focuses on achieving more energy conservation, and follows the principles of Dijkstra-Green-B to guarantee loop-free routing.

We design a link weight in two steps. First, the weight of link l is set $P_l(\bar{x}_l + x_0^v) - P_l(\bar{x}_l)$, where xl is the historical traffic volume estimation for link l. Second, we scale up the link weight by kl if link l is a trunk link. For a path p, the weight function $w_{adv}(p)$ is defined as follows:

$$w_{adv}(p) = \sum_{l \in p} \left(P_l(\bar{x}_l + x_0^v) - P_l(\bar{x}_l)\right) \cdot k_l. \qquad (8)$$

Algorithm 2. Algorithm Dijkstra-Green-Adv()

Input: $G(V, E), s, d, P, x_0^v, \bar{x}$;
Output: the advanced green path from s to d stored in $\varphi[]$;

1: **for** each node $v \in V$
2: $\quad w[v] \Leftarrow \infty; \varphi[v] \Leftarrow null$;
3: $Q \Leftarrow V; w[d] \Leftarrow 0$;
4: **while** $Q \neq \phi$
5: $\quad u \Leftarrow \text{Extract\_Min}(Q)$;
6: $\quad$ **if** $u = s$
7: $\quad\quad$ **return** $\varphi[]$;
8: $\quad$ **for** each node $v \in N(u)$
9: $\quad\quad \varpi \Leftarrow P_{(u,v)}(\bar{x}(u,v) + x_0^v) - P_{(u,v)}(\bar{x}(u,v))$;
10: $\quad\quad \varpi \Leftarrow \varpi \cdot k(u,v)$;
11: $\quad\quad$ **if** $w[u] + \varpi < w[v]$
12: $\quad\quad\quad \varphi[v] \Leftarrow u$;
13: $\quad\quad\quad w[v] \Leftarrow w[u] + \varpi$;
14: **return**;

The algorithm makes only a few modifications to the standard Dijkstra's algorithm. New inputs include the set of power-traffic functions P, the set of historical traffic volumes x, and xv 0. In the algorithm, w(v) denotes the weight of the current path from v to d and $\varphi[v]$ denotes the successor (or next hop node) node of v. N(u) denotes the set of neighbor nodes of u. The major difference between DijkstraGreen-Adv and Dijkstra is that Dijkstra-Green-Adv calculates the link weight of (u, v) in Steps 9 and 10 according to Eq. (8). The computation complexity of Dijkstra-Green-Adv is the same as that of the Dijkstra-Green-B algorithm, i.e., $O(|E| + |V|\log|V|)$ in the worst case.

Dijkstra-Green Algorithm:-

We now study the balance between green and normal QoS requirements for the routing paths. In other words, we want to investigate whether the pursuit of green may sacrifice typical routing metrics such as end-to-end delay or bandwidth etc; and how a balance can be made. As an example, we will develop an algorithm that jointly consider green and path length. Note that our previous algorithms consider path length in the sense to make the paths greener. Here the path length is considered as a separate parameter that reflects end-to-end delay.

Clearly, the green paths and the shortest paths cannot be simultaneously achieved. A typical metric to evaluate how a computed path differs from shortest path is path stretch: the ratio of the length of an s-d path to that of the shortest path between this s-d pair.

We analyze the path stretch of $w_{adv}(p)$ and find that the path stretch is small for most paths; yet there exists some big stretch when the length of the shortest path is small. Thus, we develop an algorithm which takes additional considerations for the "short" paths. Specifically, let Len(p) be the length of path p. We divide the link length by the root of the shortest path length to node d. In this way, path length will dominate in the weight for short paths, and power consumption will dominate for long paths. The weight of path $p = (s = v_0, v_1, \ldots, v_n = d)$ is defined as

$$w_g(p) = w_{adv}(p) + \sum_{i=0}^{n-1} \frac{\kappa \cdot Len(v_i, v_{i+1})}{\sqrt{Len(p_s(v_{i+1}, d))}},$$

Where $p_s(v_i, v_j)$ denotes the shortest path from node vi to node VJ, and k is a constant factor which we can use to adjust the path stretch performance. When setting k =0, the weight naturally converge to the weight in Eq. (8)

III. CONCLUSION

In this paper, we have a tendency to studied green net routing. We have a tendency to given a power model that quantifies the connection between traffic volume and power consumption. We have a tendency to valid our model mistreatment real experiments. We have a tendency to propose a hop-by-hop approach and progressively developed algorithms that guarantee loop-free routing, considerably reduce energy footprint within the internet, and together take into account QoS needs like path stretch. As a very 1st work, we have a tendency to admit that there square measure several unsolved queries. Especially, we have a tendency to have an interest in more investigating a centralized theme. This can be helpful once MPLS may be applied, and should provide theoretical bounding for the possible most power conservation.

IV. REFERENCES

[1]. M. A. Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo, "Optimal energy savings in cellular access networks," in Proc. IEEE Int. Conf. Commun. Workshops, 2009, pp. 1-5.

[2]. A. P. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, "A survey of green networking research," IEEE Commun. Surveys Tuts., vol. 14, no. 1, pp. 3- 20, Feb. 2012.

[3]. C. Gunaratne, K. J. Christensen, B. Nordman, and S. Suen, "Reducing the energy consumption of Ethernet with adaptive link rate (ALR)," IEEE Trans. Comput., vol. 57, no. 4, pp. 448-461, Apr. 2008.

[4]. P. Patel-Predd, "Update: Energy-efficient Ethernet," IEEE Spectr., vol. 45, no. 5, pp. 13-13, May 2008.

[5]. M. Gupta and S. Singh, "Greening of the Internet," in Proc. SIGCOMM, 2003, pp. 19-26.

[6]. N. Bansal, T. Kimbrel, and K. Pruhs, "Speed scaling to manage energy and temperature," J. ACM, vol. 54, no. 1, pp. 1-39, Mar. 2007.

[7]. H. Chan, W. Chan, T. W. Lam, L. Lee, K. Mak, and P. W. H. Wong, "Energy efficient online deadline scheduling," in Proc. 18th Annu. ACMSIAM Symp. Discr. Algorithms, New Orleans, LA, USA, Jan. 7-9, 2007, pp. 795-804.

[8]. M. Garrett, "Powering down," Commun. ACM, vol. 51, no. 9, pp. 42-46, 2008.

[9]. F. F. Yao, A. J. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," in Proc. 36th Annu. Symp. Found. Comput. Sci., Milwaukee, WI, USA, Oct. 23-25, 1995, pp. 374-382.

[10]. A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," in Proc. SIGCOMM, 2002, pp. 161-174.

[11]. R. G. Gallager, "A minimum delay routing algorithm using distributed computation," IEEE Trans. Commun., vol. COM-25, no. 1, pp. 73-85, Jan. 1977.

[12]. M. Andrews, A. Fernández Anta, L. Zhang, and W. Zhao, "Routing for power minimization in the speed scaling model," IEEE/ACM Trans. Netw., vol. 20, no. 1, pp. 285-294, Feb. 2012.

[13]. Y. Wang, G. Lu, and X. Li, "A study of Internet packet reordering," in Proc. Int. Conf. Inf. Netw. (ICOIN), 2004, pp. 350-359.

[14]. J. A. Aroca, A. Chatzipapas, A. Fernández Anta, and V. Mancuso, "A measurement-based

analysis of the energy consumption of data center servers," in Proc. 5th Int. Conf. Future Energy Syst. e-Energy, 2014, pp. 63-74.
[15]. T. Bingmann and D. Yordanov, "Robert Gallager's minimum delay routing algorithm using distributed computation," in Proc. Seminar (WS'07), 2007, pp. 15-31.
[16]. R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," Struct. Multidiscip. Optim., vol. 26, no. 6, pp. 369-395, 2004.
[17]. L. Zadeh, "Optimality and non-scalar-valued performance criteria," IEEE Trans. Autom. Control, vol. AC-8, no. 1, pp. 59-60, Jan. 1963.
[18]. D. Lee and N. Brownlee, "Passive measurement of one-way and two-way flow lifetimes," ACM SIGCOMM Comput. Commun. Rev., vol. 37, no. 3, pp. 17-28, 2007.
[19]. J. Charzinski, "HTTP/TCP connection and flow characteristics," Perform. Eval., vol. 42, nos. 2-3, pp. 149-162, 2000.
[20]. S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," Comput. Commun. Rev., vol. 36, no. 1, pp. 83-86, 2006