Lifetime Maximization of Green Internet by HOP by HOP Routing Approach

R.Prathiba,[1] Dr.D.Sharmila[2]

III MCA, Lord Jegannath College of Engineering & Technology

Assistant Professor/HOD, Department of Computer Applications,

Lord Jegannath College of Engineering & Technology

Abstract: we study "green" routing where we do not Prune the internet topology key observation that makes this possible is that the energy consumption for packet delivery can be different in different traffic volume. Therefore, we can select paths that consume less Power while delivery traffic. In this work the implemented hop-by hop routing for internet service with optimized power consumption. We introduced modified dijikstra green algorithm to improve the performance and gain life time of each hops, which maximize the life time of entire network.

Index Terms—energy conservation, Internet routing, hop-by-hop routing, routing algebra

INTRODUCTION:

With the widespread use of the Internet, energy conservation has become a major concern in networks over the past years. Current networks are not designed with energy optimization as an objective or a constraint. They are often over-provisioned and designed for peak traffic. These design strategies give us huge potential for energy saving in current networks ENERGY conservation has become a global concern nowadays and energy cost is predicted to increase in the forthcoming future.

The network components to be turned off are carefully chosen and tradeoffs are investigated to balance network performance and energy conservation. To realize these approaches, MPLS or additional protocols are usually necessary. We study "green" routing where we do not prune the Internet topology. A key observation that makes this possible is that the energy consumption for packet delivery can be different in different traffic volumes. Therefore, we can select paths that consume less power while delivering traffic [5][6] and adaptive link rate[1][2].

RELATED WORK:

Only just there are moving parts focusing on energy saving at the network-broad level such as manipulative energy saving network routing protocols based on energy-aware network devices. Gupta et al.[3] firstly proposed importance of considering energy saving from a network protocol [7][4] they have studied the global energy consumption minimization routing problem considering the speed scaling model and the powering down model under static traffic developed [6] GreenTE, an energy awaken traffic engineering scheme by putting links into sleep mode while obeying the performance constraints. [8]Proposed an online energy aware traffic production scheme based on source steering and they planned energy critical path as steering selection basis to achieve more energy saving [9]. Avallone et al. [8] designed a joint admission control and routing scheme in order to minimize active network devices while meeting the performance requirement. Some other network energy-efficient heuristic algorithms were proposed in[11][1O]. However, most existing works use centralized strategies or source routing for routing path assignment, they usually go to a solver to get results or develop related heuristics strategies. These methods are not straightforward to realize especially in large distributed networks because of the pointlessness of obtaining global information and high computation cost. They are also not flexible for packet to adjust path selection during routing according to the dynamic condition of links. With the scale of networks keeping upward, compared with federal strategy, a hop-by-hop distributed scheme is a more scalable and flexible choice. But how to achieve the same or near the same global energy optimization in a distributed way brings in some challenges. First of all, without knowing the whole network topology and traffic demands, each node makes their own decision of choosing working mode/speed and selecting which neighbor to forward a packet. Lacking global information, it is important to make sure that the individual operation will benefit the global energy saving performance rather than impair it. Another serious issue is that a spread scheme should work correctly which means it is loop-free and can converge well without oscillations. Additionally, only considering energy saving will sacrifice the network performance such as bringing in congestion and intolerable delay. If increasing traffic combined into the link with low energy cost, it will cause congestion and degrade the network performance. So how to deal with the tradeoff between performance such as delay and energy efficiency is also necessary. The problem of multi-metric QoS routing have been widely investigated in the past years(e.g to cite a few). However, to the best of our knowledge, they did not consider energy as a metric and did not try fully spread hop-by-hop strategy which can approach global optimal. We develop a spread hop-by-hop energy efficient steering scheme that concurrently minimizes the global energy cost and the total traffic delay. We propose to leverage speed scaling as the hardware level support for energy saving. It dynamically adjusts network devices' working speed according to the traffic load. Using speed scaling, we study how to direct traffic demands to flow through energy saving paths while optimizing its delay in a distributed manner. In order to maintain loop-freeness we propose a spread multi-path finding algorithm as the preprocess of the routing. It provides nodes with next-hop-choice information without causing loops during routing. Our multi-path routing guarantees better load balancing compared with existing single loop-free path routing methods. Also, it has little communication among nodes and is easy to implement. By applying and extending the optimization theory in[12] , our distributed scheme can quickly converge to the near global optimal with respect to energy saving and traffic delay. The main charities of this paper are the following: First, we celebrate the minimizing-energy spending and delay problem in wired networks in a multi-criterion optimization framework that simultaneously optimizes the two objectives. Combining the two objectives by secularization, we can get the Pareto optimal curve which is the optimal tradeoff between energy saving and packet delay. Second, based on the formalization of the problem, we develop a distributed loop-free scheme to route traffic demands. Experiment results show that our algorithm can converge quickly and achieve near global optimal value. Third, we consider dynamic traffic with bursts and develop a distributed algorithm to adapt to the dynamic change of the traffic. We show through experiments that our algorithm works rather well under dynamic traffic in comparison with theoretical optimal value.

GREEN INTERNET MODEL:

We model a network topology as an undirected graph with bi-directional links $G = (N,E)$, N is the set of all nodes and E is the set of all links in the network topology. We assume that the traffic is injected into the network from source node i to destination node j with the expected rate rij . Our goal is to route traffic on the network aiming at optimizing network performance measured as network energy consumption and traffic delay. We assume that fractional routing is permitted which means that a routing demand can be splitted on multiple paths [7] .(Since the packet reordering problem has been widely studied[12] , we will not discuss it here.) Here, we use the same definition of variables as in. Let $ti\ j$ represent the total traffic from node i destined to node j. $ti\ j$ is the sum of rij and the traffic flowing to j through node i from other nodes. Let Ni denote the set of neighbors of node i. The routing variable $\phi i\ jk$ represents the proportion of traffic tij allocated to edge (i, k), where node k is one of the neighbors of node i. From flow conservation, we have

$Ti\ j = rij + \Sigma\ k\in Ni\ tkj \times \phi kj\ i.$ (1)

Let xik be the total expected traffic passing through link (i, k).Then, $xik = \Sigma\ j\in N\ ti\ j\times \phi i\ jk.$ (2)

Let $CAik$ represents the capacity of link (i, k), the flow of each link cannot exceed the capacity of that link. That is, $0 \le xik \le CAik$. (3) The routing variable $\phi i\ jk$ is nonnegative and satisfies the following constraints.

1. $\phi i\ jk = 0$ if $(i, k) \notin E$ or if $i = j$.
2. $\Sigma k\in Ni\ \phi i\ jk = 1$.

The routing algorithm chooses paths for packets based on the weight of links. We define the weight of each link by a new cost function simultaneously considering energy cost and delay. We aim at minimize equally two metrics at the network broad level which is a bi-objective optimization. The energy function $fik(xik)$ represents the energy cost of link (i, k) when transmitting xik units of traffic through it.(It actually contains energy cost of the link and related devices when transmitting packets through them.) The delay function $dik(xik)$ is the delay of transmitting xik units of traffic on the link (i, k). Both two functions are computed from the expected traffic. By summing up the energy cost of each link, we define FT as the total energy cost of the network

as $FT = \Sigma\ (i;k)\in E\ fik(xik)$. And the total traffic delay DT in the network is formulated as $DT = \Sigma\ (i;k)\in E\ dik(xik)$.

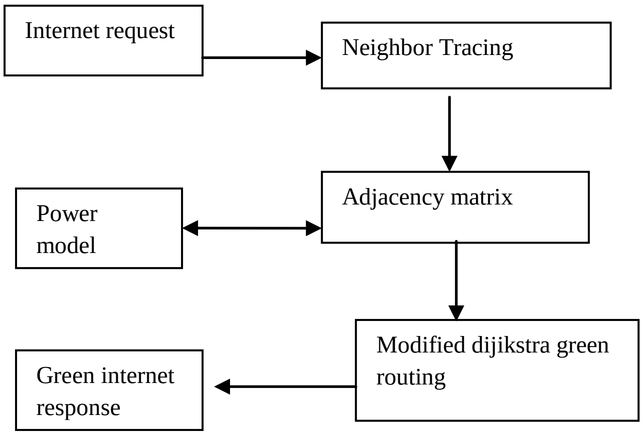
We merge two objectives together in a scalarization way using weighted sum method. Weighted sum method provides the sufficient condition to generate optimal points in multi-objective optimization. We call these optimal points Pareto Optimal [13]. We give each objective function a weight to represent the preference on the objective. We define the final single objective CT to be the total cost of the network as $CT = w1 \times FT + w2 \times DT$. The sum of the weights of each objective equals to 1 as in and both $w1$ and $w2$ are nonnegative. $w1 + w2 = 1$. A $(w1,w2)$ pair corresponds to a Pareto Optimal point. With many $(w1,w2)$ pairs, we can get a set of Pareto Optimal points. We formalize this bi-objective optimization in the following way. minimize CT is to find the Pareto best points for minimize FT and DT .(P) Min CT subject to

$xik = \Sigma\ j\in N\ ti\ j \times \phi i\ jk\ \forall i, k\ ti\ j = rij + \Sigma\ k\in Ni\ tkj \times \phi kj\ I\ \forall i, j$ (flow conservation)

$\Sigma\ k\in Ni\ \phi i\ jk = 1\ \forall i, j$ (routing variable restriction)

$0 \le \phi i\ jk \le 1\ \forall i, k, j$ (capacity restriction) $xik \le CAik\ \forall i, k$ (routing variable restriction) When the traffic matrix is fixed, this optimization problem (6) can be solved by the optimization solver in a centralized way. However, centralized method needs the whole network topology and traffic matrix as the input and takes time to compute routes for all traffic demands and it may highly rely on central management and scheduling. To apply this compound calculation is pretty hard in particular in large spread networks, it is therefore meaningful to realize such a bi-objective steering problem in a spread manner. In next section, we will present a spread steering scheme to solve this problem.

ARCHITECTURE MODEL



DIJIKISTRA GREEN ALGORITHM:

We now study hop-by-hop green steering (Green-HR). We first propose a path weight and a baseline algorithm Dijikstra-Green-B to achieve loop-free. We then study some intrinsic relationships between link weights and power spending, and develop an advanced algorithm Dijikstra- Green-Adv that improves energy conservation. We additional develop algorithm Dijikstra-Green that parallel considers energy conservation and path stretch. From Section 4, we see that the key is to develop an appropriate weight for a path so that it incorporates "green" and holds isotonicity. A Dijikstra-oriented algorithm can then be urbanized 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. Then, with the virtual traffic volume, the link power is computed following the power function in Section 3. Formally, for each destination node *d*, we assign $xv0$ as the start traffic volume. This $xv0$ is determined by the total bandwidth associated with *d* and we will specify this later. For a path

$p = (s = v0, \ldots, vn = d)$, the virtual traffic volume for each

link $l = (vi, vi+1)$ $i \in \{0, 1, \ldots, n-1\}$ is set to $xvl = xv0 \cdot \beta h$.

Here β ($\beta > 1$) is a constant and *h* is the hop number of the *lightest-shortest* path $pls(vi+1, d)$. Here, the lightest shortest path $pls(vi+1, d)$ means the path from $vi+1$ to *d* that has the lightest path weight and the least number of hops. The weight of link *l* is set to $Pl(xvl)$, where $Pl(\cdot)$

follows the power function in Section 3. The weight of path $p = (s = v0, v1, \ldots, vn = d)$ is the sum of the weight of each link:

$wb(p) = n\Sigma{-1}\ i{=}0\ P(vi;vi+1)\ (xv0 \cdot \beta \mathrm{Hops}(pls(vi+1;d)))$

Here, function Hops(p) returns the hop number of path *p*. Note that a link's weight is not a static value, and may vary with path *p* and destination node *d*. However, we can prove the strict left-isotonicity of this path weight structure. We define an algebra $(S, \oplus \preccurlyeq, wb)$ based on Eq. (6)

where *S* is $R+$, $\preccurlyeq$ is $\leq$, *wb* is given in Eq. (6), and $wb(p) \oplus wb(q)$ is equal to $wb(p \circ q)$ which can be calculatedby Eq. (6).

Theorem 5.1. The algebra $(S, \oplus \preccurlyeq, wb)$ defined by Eq. (6) is strictly left-isotonic.

Proof: As shown in Fig. 4, assume $p1$ and $p2$ are two

paths from node s to node d. Without losing generality, Assume that $p1$ is lighter than $p2$, i.e., $wb(p1) \prec wb(p2)$. We check the order relation between $wb(q \circ p1)$ and $wb(q \circ p2)$, i.e., the weights after concatenating $p1$ and $p2$ to path q,
respectively.
v d 0 $p1$ $p2$ q $v1$ vn=s
The lightest-shortest
path from vi+1 to d vi+1
Fig. 4: The topology used to prove the strict left-isotonicity of the path weight structure defined by Eq. (6). Assume $q = (v0, v1, v2, \ldots, vn = s)$. According to Eq. (6)
we have
$wb(q \circ p1) = n\Sigma{-1}\ i{=}0$
$P(vi;vi{+}1)\ (xv0\ \beta \text{Hops}(pls(vi{+}1;d))) + wb(p1)$
and
$wb(q \circ p2) = n\Sigma{-1}\ i{=}0\ P(vi;vi{+}1)(\ xv0 \cdot$
$\beta \text{Hops}(pls(vi{+}1;d))) + wb(p2)$.
Note that the length of the lightest-shortest path from vi to d is independent of $p1$ or $p2$. Because $wb(p1) \prec wb(p2)$ means $wb(p1) < wb(p2)$, we can obtain from the above equations $wb(q \circ p1) < w(q \circ p2)$, which means $wb(q \circ p1) \prec wb(q \circ p2)$. This implies that the strict left-isotonicity holds, and completes the proof. Based on Theorem 4.1, 4.2 and 5.1, we can achieve a consistent (thus loop-free) hop-by-hop routing by applying a Dijikstra-like algorithm. We develop Algorithm Dijikstra-Green-B. ***P*** in the inputs denotes the set of the power-traffic functions of all the links in E. 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) of v. $N(u)$ denotes the set of neighbor nodes of u. $h[u]$ is used to store the hop number of the lightest-shortest path from u to d. There are a few differences between Dijikstra-Green-B and the standard Dijikstra. 1) A sink tree rooted at d is calculated and the algorithm halts once s is extracted (Step 5 to 7). 2) $h[u]$ is used to record the lightest-shortest path. 3) A link weight is calculated according to Eq. (6) in Step 9 and 10.

Algorithm Dijikstra-Green-B()

Input: $G(V,E)$, s, d, ***P***, $xv0$, β;

Output: the green path from s to d which is stored in $\varphi[]$;

1: **for** each node $v \in V$
2: $w[v] \Leftarrow \infty$; $\varphi[v] \Leftarrow null$; $h[v] \Leftarrow \infty$;
3: $Q \Leftarrow V$; $w[d] \Leftarrow 0$; $h[d] \Leftarrow 0$;
4: **while** $Q \neq \phi$
5: $u \Leftarrow$Extract Min(Q);
6: **if** $u = s$
7: **return** $\varphi[]$;
8: **for** each node $v \in N(u)$
9: $x \Leftarrow xv0$
$\cdot\ \beta h[u]$;
10: $\varpi \Leftarrow P(v;u)(x)$;
11: **if** $w[u] + \varpi < w[v]$
12: $\varphi[v] \Leftarrow u$;
13: $w[v] \Leftarrow w[u] + \varpi$; $h[v] \Leftarrow h[u] + 1$;
14: **else if** $w[u] + \varpi = w[v]$ **and** $h[u] + 1 < h[v]$
15: $\varphi[v] \Leftarrow u$; $h[v] \Leftarrow h[u] + 1$;
16: **return** null; //d is unreachable

The computation complexity of Dijikstra-Green-B is the same as that of the standard Dijikstra 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 then best case is $O(1)$. Note that Dijikstra-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 Dijikstra.

PERFORMANCE:

Now we will discuss and analyze a few issues that are important to hop-by-hop green routing. First, we discuss the bounds on the the optimal power saving without topology pruning, and the power saving ratio that Green-HR can achieve. Also, we compare the power saving ratio of Green- HR with that of topology pruning approaches. Second, we discuss the routing dynamics of Green-HR, and show that routing oscillations and transient micro-loops can be avoided. Third, we study the relationship between Green- HR and QoS requirements, and show that it is impossible to find a strictly left-isotonic path weight structure optimizing one path weight while bounding another, due to the intrinsic nature of routing algebra.

6.1 Power Saving Ratio of Green-HR

The hop-by-hop green routing approach saves energy by leveraging trunk links, without any topology pruning. The problem of maximizing the power saving with trunk links using MPLS-like routing has been shown to be NP-hard [18]. Green-HR uses heuristics to solve the problem, and our main concern is to develop an OSPF-like routing which requires loop-free hop-by-hop forwarding and Dijkstra oriented computing. Thus, there is no guarantee on the optimality of power saving. However, we can still do some analysis. Without topology pruning, the power saving that can be achieved is related to the traffic load. In the case when the network is heavy-loaded, all links consume the maximum power and no power can be saved. Clearly, Green-HR will not induce more power consumption in this case. In the case when the traffic is extremely small, each trunk link has only one physical link powered on. Green-HR can achieve such a power saving even when the traffic is larger, because our path weights tends to put less traffic on the trunk links. These two cases show the lower and the upper bounds of the power saving. Formally, we consider a simplified model. Without loss of generality, let λ be the ratio of the trunk link number to the total link number. Each trunk link consists of m parallel physical links and each physical link consumes the same power. The upper bound of optimal power saving ratio without topology pruning can be presented by

$\lambda|E|(m-1)$

$\lambda|E|m + (\lambda - 1)|E| =$

$\lambda(m-1)$

$\lambda m + \lambda - 1$

. (10)

Generally, a larger λ and/or m results in more power saving. In an extreme case when all links are trunk links, i.e., $\lambda = 1$, the upper bound is $m-1$ m . We consider the relation between Green-HR and topology pruning approaches. Even after some links or nodes are pruned, Green-HR can still be used on the remaining topology for further energy conservation. However, it is interesting to ask whether Green-HR saves more energy or less than topology pruning approaches. We do a comparison using the simplified model above. Let σ be the average number of powered on physical links in a trunk link with Green-HR ($\sigma \leq m$), and σ' be the number in the pruned topology ($\sigma < \sigma' \leq m$). We can derive the condition under which Green-HR can save more power:

$\lambda|E| > \sigma'\sigma' - \sigma\,(|E| - |V| + 1)$. (11)

Generally, in networks that have a big λ and a small σ/σ', Green-HR may outperform topology pruning approaches. In the Internet, trunk links are deployed more and more, which is a common method to upgrade the network capacity.

CONCLUSION:

We consider a routing problem simultaneously optimizing energy consumption and delay internet. We formulate the problem as a bi-objective optimization problem. We then propose a hop-by-hop distributed loop-free routing scheme to solve this bi-objective optimization problem. By obeying the optimal routing law, our scheme can approach the global optimal. Experiments have been carried out using real network data and the results show that, after several rounds of iteration, the distributed algorithm converges to the near Pareto optimal.

REFERENCE:

1. R. Kubo, J. Kani, H. Ujikawa, T. Sakamoto, Y. Fujimoto, N. Yoshimoto,and H. Hadama, "Study and Demonstration of Sleep and Adaptive Link Rate Control Mechanisms for Energy Efficient 10GEPON," *IEEE/OSA JOCN*, vol. 2, pp. 716–729, 2010.
2. C. Gunaratne and K. Christensen, "Ethernet Adaptive Link Rate: System Design and Performance

Evaluation," in *Proc. of the 31st IEEE Conference on Local Computer Networks*, 2006, pp. 28–35

3. A. P. Bianzino, L. Chiaraviglio, and M. Mellia, "Distributed Algorithms for Green IP Networks," in *IEEE INFOCOM Workshop on Green Networking and Smart Grid*, March 2012.
4. J. Carter and K. Rajamani, "Designing Energy-Efficient Servers and Data Centers," *Computer*, vol. 43, pp. 76–78, 2010.
5. W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks: reducing energy consumption by shutting off cables in bundled links," in *Proc. of the first ACM SIGCOMM workshop onGreen networking*, 2010, pp. 29–34..
6. G. Lin, S. Soh, K. Chin, and M. Lazarescu, "Efficient heuristics for energy-aware routing in networks with bundled links," *Computer Networks*, 2013, vol. 57(8), pp. 1774–1788.
7. V. Siddhartha, P.V. Ramakrishna, T. Geetha and A. Sivasubramaniam, Automatic generation of energy conservation measures in buildings using genetic algorithms," *Energy and Buildings*, vol. 43, pp. 2718–2726, 2011.
8. M. Xu, Y. Shang, D. Li, X. Wang. "Greening data center networks with throughput-guaranteed power-aware routing," *Computer Networks*, Vol. 57(15), pp. 2880–2899, 2013.
9. C. Hou, F. Zhang, A. F. Anta, L. Wang, and Z. Liu. "A Hopby- hop Energy Efficient Distributed Routing Scheme," in *Proc. Of Greenmetrics Workshop*, 2013.
10. A. Cianfrani, V. Eramo, M. Listanti, and M. Polverini, "An OSPF enhancement for energy saving in IP networks," in *IEEE INFOCOM Workshop on Green Communications and Networking*, April 2011, pp.325–330.
11. Q. Li, M. Xu, Y. Yang, L. Gao, Y. Cui, and J. Wu. "Safe and Practical Energy-Efficient Detour Routing in IP Networks," *IEEE/ACM Transactions on Networking*, vol. 22(6), pp.1925–1937, 2014.
12. F. Cuomo, A. Abbagnale, A. Cianfrani, and M. Polverini, "Keeping the Connectivity and Saving the Energy in the Internet," in *IEEE INFOCOM Workshop on GCN*, April 2011.
13. E. Amaldia, A. Caponea, L. Gianoli. "Energy-aware IP traffic engineering with shortest path routing. Computer Networks," *Computer Networks*, vol. 57(6), pp.1503–1517, 2013