106117057

106117079

AN ENERGY SAVING ALGORITHM FOR A MODIFIED GREEN OSPF PROTOCOL (MODIFIED)

Abstract

In our paper we tackle the problem of energy saving in IP networks. We try to come up with a strategy based on a modification of current link-state routing protocols, such as OSPF. By this method, IP routers are given the ability to power off some network links during low traffic periods. The given solution has 3-phases: in the first phase some routers are selected as exporter of their own Shortest Path Trees (SPTs); then, one the neighbors of these routers run a modified Dijkstra algorithm to find the links to power off; finally, new network paths on a modified network topology are calculated. Studies show that, in a network, more than the 60% of links can be switched off. In the second phase, exporters are chosen based on parameters such as the connectivity of the routers and number of packets processed by the router.

Introduction

1. OSPF Protocol

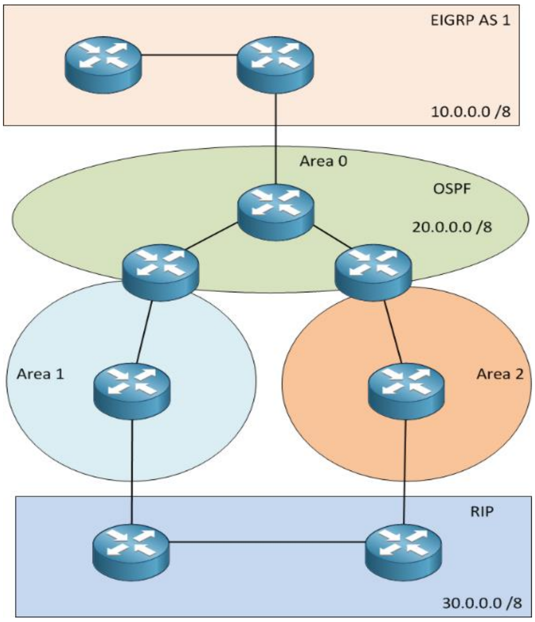
Open Shortest Path First (OSPF) is one of the routing protocols, which uses link-state routing which is used to find the optimal path between the source and the destination router using its own Shortest Path First). OSPF is developed by Internet Engineering Task Force (IETF) as one of the Interior Gateway Protocol (IGP), i.e., the protocol which aims at moving the packet within a large autonomous system or routing domain. Open Shortest Path First (OSPF) was designed as an interior gateway protocol (IGP), for use in an autonomous system such as a local area network (LAN)s. OSPF algorithm allows every router to calculate the cost of the routes to any destination.

2. OSPF Protocol working / Algorithm

OSPF keeps link state databases, on every router on which it is implemented. A router i/f will send its link cost to its neighbouring routers through multicasting, called the hello procedure. Routers keep sending their hello packets, and changes in the cost of their links are known to their neighbouring routers. The process of flooding link state information is known as synchronisation. All routers continuously update their link state databases with information about the network and update the routing tables. OSPF uses its own Shortest Path Algorithm in order to route the packets through the network. To know about the network topology, it uses link state routing in which each router shares the knowledge of its neighbourhood with every other router in the network.

Initialization

```
N = {A}      // A is a root node.
for all nodes v
if v adjacent to A
then D(v) = c(A,v)
else D(v) = infinity
loop
find w not in N such that D(w)= min.
Add w to N
Update D(v) for all v =adj(w),
       v not in N:
D(v) = min(D(v) , D(w) + c(w,v))
Until all nodes in N
```



3. Motivation

Internet is very fast growing nowadays. This leads to an exponential increase in power required to operate the entire global infrastructure every year. One of the most used routing protocols is OSPF, a link-state routing protocol. It consumes a lot of energy and memory for routing the packets in the corresponding network. This accounts for nearly 5% of the power required to run the internet. This percentage of energy consumption is too high on a global level. This is an unsustainable condition, which will affect the future generations. So, we aim to reduce the power consumption of the OSPF protocol and the memory consumed by the routers using OSPF protocol.

4. Contribution

We made a few modifications to the existing OSPF protocol. We try to power down a number of links and routers during low traffic periods, in order to reduce the power consumption of each router. We try to select a few routers called exporters, based on which we power down some of the routers. The function of the exporters is to exchange the information about the network to other routers, so that other routers get information about the network. The selection of routers is based on a specific criterion and not randomly in order to minimise the number of routers that send their information so that power consumption is reduced.

5. Paper Organisation

The first section gives an introduction about the overall work that is done. The introduction section gives details about the default method, motivation and contribution. The second section gives details about the related work that was carried out in the past by researchers and summary of their works. It also includes the limitations of the previous works and how we addressed these limitations. The third section is dedicated to proposed work. It includes working of the proposed method, the algorithm and the explanation of the algorithm. The rest of the paper consists of simulations and the results. The final section concludes the paper followed by references.

Related Work

1. Previous Work carried out

a. Quality of service (QoS) sensitivity for the OSPF protocol in the airborne networking environment

- examines the use of the open shortest path first (OSPF) protocol for use within the future airborne network.

b. The Research on the OSPF Security Optimizing of Campus Network

- The paper is aimed at the OSPF security concerns in large networks and research the OSPF security optimizing design from the accessibility of the non-backbone area route.

c. Robust optimization of OSPF/IS-IS weights

- The paper's objective is to route requests along an OSPF based network in order to avoid congestion exceeding capacities with resulting packet loss and back-off in TCP.

d. OSPF Monitoring: Architecture, Design and Deployment Experience

- Describes the architecture, design and deployment of a monitoring system for OSPF, an IP intradomain routing protocol in wide use.

e. An Alternative Genetic Algorithm to Optimize OSPF Weights

- Here, a strategy based on a genetic algorithm (GA), is presented to optimize weights for OSPF routing. It can be seen as an alternative to the local-search method.

f. Improving Convergence Speed and Scalability in OSPF

- In this paper, performance analysis of the Wireless and Wired computer networks for combination of conventional model of RIP and OSPF is evaluated through simulation.

g. A Genetic Algorithm for the Weight Setting Problem in OSPF Routing

- OSPFWS searches for a set of weights that improves network performance. We study the problem of optimizing OSPF weights with the objective of minimizing network congestion.

h. A biased random-key genetic algorithm for OSPF and DEFT routing to minimize network congestion

- Presents a biased random-key genetic algorithm for WSP using both protocols.

i. A Performance Comparison of MD5 Authenticated Routing Traffic with EIGRP, RIPv2, and OSPF

- A network model of four Cisco routers has been employed with an ON/OFF traffic model used to describe text files transmissions over the network.

j. Survivable composite-link IP network design with OSPF routing

- The paper tries to find a set of OSPF weights that optimizes network cost with respect to single arc failures. It proposes an algorithm to find optimal solutions for this problem.

2. Summary of the Works

Most of the research papers mentioned above doesn't give the return of investment by changing their routers and their protocols. Few of them compromised the network coverage for efficiency. Some implementations don't yield the expected results for the trade-offs that were made. Some of the papers don't consider the high memory usage that is required by the OSPF protocol. Bad design results in high CPU utilization. Flooding of LSA every 30 minutes creates extra traffic and brings network utilization up.

3. Overcoming the limitations

We try to reduce the memory usage and the power consumption in OSPF by not compromising on the network coverage and efficiency, in order to this we try to power down a number of links and routers during low traffic periods, in order to reduce the power consumption of each router. We try to select a few routers called exporters, based on which we power down some of the routers. The function of the exporters is to exchange the information about the network to other routers, so that other routers get information about the network. The exporters selection helps to reduce the memory consumption by the entire network as only selected routers transport their routing table and also helps us to reduce the power consumption.

Proposed Work

1. Working of proposed method (EAR Algorithm)

In this section the EAR algorithm that is the objective of our study is presented in detail. The purpose of EAR algorithm is to minimize the active links of a network, i.e. the number of links to be used to route traffic; this goal is achieved by forcing a subset of routers to assume routes different from those indicated in their SPTs. The set of network routers is divided into three subsets: exporter, importer and neutral routers. An Exporter Router (ER) is associated to a number of Importers Routers (IRs), whereas an IR is associated to a single ER. An IR will compute its routing path tree taking into account the SPT evaluated by the corresponding ER. Neutral Routers (NRs) act as in the classical OSPF routing protocol.

2. Proposed Algorithm

Let G (*V, E, W*) denote the weighted graph of the network where V is the set of nodes, E is the set of unidirectional links and W is the set of weights associated to each link. Let SPG (V, Es, Ws) be the subgraph of G obtained by the superposition of SPTk (k=1,··, R), i.e. SPG=⋃i=1··RSPTi. Es identifies all the links used to route packets, in the following we indicate this class of links as *active links*. Let LD be the cardinality of Es:

Pseudo Code

1. LD=‖ES‖=‖⋃i=1··RSPTi‖
2. if Ld lesser than or equal to L.
3. if Ld = L then link costs are identical else Ld < L
4. Switch off L – Ld links belonging to set E – Es.
5. Min (Ld) = 2 (R-1) = Lmin , R = no of routers
6. Lmin = min no of links that ensures network connectivity.
7. Lmin < Le < Ld < L, Le = no of links to route traffic when EAR is done .
8. ne = (Ld - Le)/(Ld-Lmin), ne = max no of links that can be switched off.
9. Select exporters E ⊂ R,
10. If(packetsProcessed (E)==maximum && degree(E)==maximum)
11. Switch of routers Rsw ⊂ R
12. If(Rsw!= E && Rsw present in (L - Ld))
13. Run Dijikstra for all the exporters and share the routing table.

3. Explanation about the Algorithm

The number of ERs, that we denote with Re, and how they are selected strongly affect the performance of EAR algorithm. Let's consider an IR router (rn) that is a neighbour of an ER (re). The SPT (re) is assumed by rn as its routing tree. In other words, the routing tree of rn is topologically identical to SPT (re), but having rn as root node. We indicate with MPT (rn,re) this modified version of the SPT(re). In order to achieve this result SPT (re) can be divided in two portions: i) the subtree having rn as root node and ii) the rest of the tree. The time complexity of the proposed algorithm is same as Dijkstra since it is based on shortest path first routing.

Complexity

- Worst case time complexity : **O (E + V.log(V))**
- Average case time complexity : **O (E + V.log(V))**
- Best case time complexity : **O (E + V.log(V))**
- Space complexity : **O (V)**

Simulation Results and analysis

1. Simulation Setup

The program was run in a gcc compiler taking a graph of a network as an input. The graph was given as a cost matrix considering it as a directional graph. We first get the no of nodes in the network from the user as an input along with the connections or edges which are mentioned in the cost matrix. We calculate the MPT for each and every node considering it as source and display the MPT as a graph. The efficiency of the setup is measured as the percentage of routers that can be switched off and is given as a numerical percentage to the user. Note – the exporters are not chosen randomly.

2. Simulation Parameters

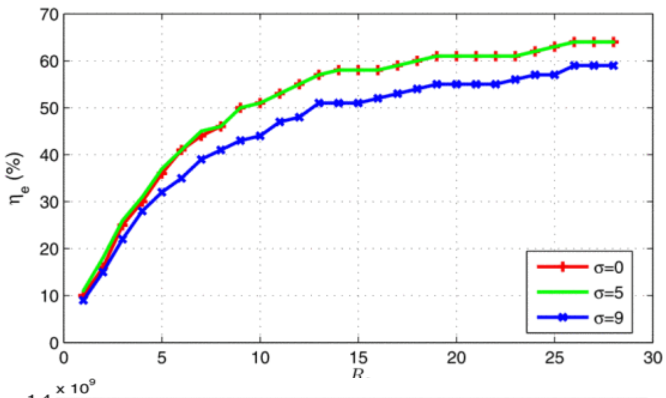
We have considered ne as the performance index to evaluate the EAR algorithm, and the percentage of links that EAR algorithm allows to power off, i.e. (LD−Le), with respect to the maximum number of links that could be powered off, i.e. (LD−Lmin). We do not refer to the overall network links because only a subset of links is used to route traffic even when the Dijkstra algorithm is performed; this subset of links is composed by links belonging to at least one SPT.

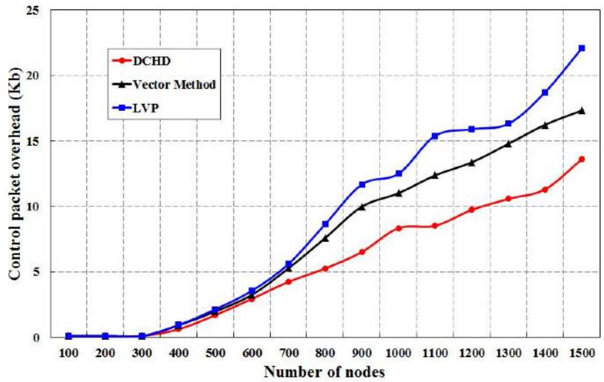
Parameter Table

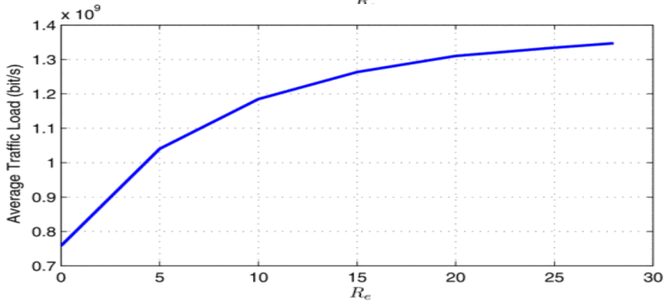
Parameter	Explanation
Ex	Exporter chosen based in degree and packets processed
Ld	No of links that remain active after EAR , optimum(Ld)
ne	Max no of links that could be powered off
ATL	Avg traffic load

3. Simulation Results and Analysis

We have considered real network topologies to evaluate EAR algorithm performance. We have fixed the link weights by assuming a uniform distribution among a mean value equal to 10, with a given standard deviation σ. In particular, we have considered three different scenarios: a constant cost case (σ=0) and two variable cost ones, characterized by σ=5 and σ=9, respectively.







The above graph shows that increasing Re., and so decreasing the number of active links, leads to increasing average traffic load on active links. In particular passing from the Dijkstra algorithm, Re=0, to the EAR algorithm with the maximum energy saving, Re=28, the average traffic load goes up of about 100.

OUTPUT

```
1: 2(10.00) 3(12.00) 4(32.00) 5(18.00) 6(23.00)Edges :1 4
Edges :3 5
Edges :0 1
Edges :2 5
Edges :0 2
src 0
0 1 1 0 0 0
1 0 0 0 1 0
1 0 0 0 0 1
0 0 0 0 0 1
0 1 0 0 0 0
0 0 1 1 0 0

2: 1(10.00) 3(20.00) 4(30.00) 5(8.00) 6(21.00)Edges :1 4
Edges :3 5
Edges :0 1
Edges :2 5
Edges :0 2
src 0
0 1 1 0 0 0
1 0 0 0 1 0
1 0 0 0 0 1
0 0 0 0 0 1
0 1 0 0 0 0
0 0 1 1 0 0

3: 1(12.00) 2(20.00) 4(20.00) 5(15.00) 6(11.00)Edges :1 4
Edges :3 5
Edges :0 1
Edges :2 5
Edges :0 2
src 0
0 1 1 0 0 0
1 0 0 0 1 0
1 0 0 0 0 1
0 0 0 0 0 1
0 1 0 0 0 0
0 0 1 1 0 0

4: 1(32.00) 2(30.00) 3(20.00) 5(22.00) 6(9.00)Edges :1 4
Edges :3 5
Edges :0 1
Edges :2 5
Edges :0 2
src 0
0 1 1 0 0 0
1 0 0 0 1 0
1 0 0 0 0 1
0 0 0 0 0 1
0 1 0 0 0 0
```

```
5: 1(18.00) 2(8.00) 3(15.00) 4(22.00) 6(13.00)Edges :1 4
Edges :3 5
Edges :0 1
Edges :2 5
Edges :0 2
src 0
0 1 1 0 0 0
1 0 0 0 1 0
1 0 0 0 0 1
0 0 0 0 0 1
0 1 0 0 0 0
0 0 1 1 0 0

6: 1(23.00) 2(21.00) 3(11.00) 4(9.00) 5(13.00)Edges :1 4
Edges :3 5
Edges :0 1
Edges :2 5
Edges :0 2
src 0
0 1 1 0 0 0
1 0 0 0 1 0
1 0 0 0 0 1
0 0 0 0 0 1
0 1 0 0 0 0
0 0 1 1 0 0


The final matrix is :
0 1 1 0 0 0
1 0 0 0 1 0
1 0 0 0 0 1
0 0 0 0 0 1
0 1 0 0 0 0
0 0 1 1 0 0
The links that are needed to be switched off :
0 3

1 2

2 4


4 5


44.44% is the percentage links switched off
The exporters are :2 0
17 9 14 5 4 8
Router 4 is Switched off temp
```

Conclusion

We have analysed the problem of energy efficiency in link state routing. We propose an ideal network-level strategy, i.e. a solution based on synchronization among the routers in the network, to save energy during low traffic periods or low traffic time, i.e. night hours. In this way it is possible to deflect the packet traffic and to power off some network links. In a real network scenario, we shown that EAR algorithm allows to power off even more than the 60% of possible network links; we also performed an analysis on algorithm impact on links load, showing that EAR is able to uniformly distribute the load increase on active links. Finally, our algorithm works on most cases considering the cost between the routers and the number of packets processed by each router in order to narrow down to some particular or ideal number of routers as exporters in order to reduce the memory consumption and power consumption in the network.

References

1. J. G. Koomey, "Estimating total power consumption by servers in the U.S. and the world" in Lawrence Berkeley National Laboratory and Consulting Professor, Stanford University, Feb. 2007.
2. M. Gupta, S. Singh, "Greening of the Internet", *Proceedings of ACM SIGCOMM 2003*, August 2003.
3. K. Christensen, C. Gunaratne, B. Nordman, "Managing Energy Consumption Costs in Desktop PCs and LAN Switches with Proxying Split TCP Connections and Scaling of Link Speed", *Int. Journal of Network Management*, vol. 15, no. 5, pp. 297-310, September/October 2005.
4. M. Gupta, S. Singh, "Using Low-power Modes for Energy Conservation in Ethernet LANs", *INFOCOM*, 2007.
5. L. Chiaraviglio, M. Mellia, F. Neri, "Reducing Power Consumption in Backbone Networks" in IEEE ICC, Dresden, Germany, 2009.
6. L. Chiaraviglio, M. Mellia, F. Neri, "Energy-aware Backbone Networks: a Case Study" in IEEE ICC, Dresden, Germany, 2009.
7. A. Nucci, A. Sridharan, N. Taft, "The Problem of Synthetically Generating IP Traffic Matrices: Initial Recommendations", *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 2, pp. 19-32, July 2005.
8. B. Fortz and M. Thorup, "Increasing internet capacity using local search. Technical Report IS-MG 2000/21", Universit´e Libre de Bruxelles, 2000. http://smg.ulb.ac.be/Preprints/Fortz00 21.html.
9. B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights". In Proc. 19th IEEE Conf. on Computer Communications (INFOCOM), pages 519–528, 2000.
10. B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world". IEEE Journal on Selected Areas in Communications, 20(4):756–767, 200
11. J. T. Moy. OSPF version 2, "Network Working Group", http://search. ietf.org/rfc/rfc1247.txt, July 1991.
12. Cisco. Configuring OSPF, 1997. Documentation at http://www.cisco.com/ univercd/cc/td/doc/product/software/ios113ed/113ed cr/np1 c/1cospf.htm.
13. A. Shaikh et al., “A Case Study of OSPF Behavior in a Large Enterprise Network,” in Proc. ACM SIGCOMM Internet Measurement Workshop (IMW), November 2002.
14. M. Ericsson, M.G.C Resende, and P.M. Pardalos. "A genetic algorithm for the weight setting problem in OSPF routing", 2001. To appear in J. Combinatorial Optimization in 2002
15. D. Staehle, S. Koehler, U. Kohlhaas, "Optimization of IP Routing by Link Cost Specification", University of Wuerzburg, 2000.