Optimized and Adaptive Link State Routing Strategy

Avishek Anand,
Indian Institute Of Information Technology, Allahabad.
avishekanand@iiita.ac.in

Kiran R.K,
BMS College of Engineering, Bangalore.
kiran_r_k83@yahoo.co.in

Abstract

The dominant link state protocols like OSPF despite their advantages require the flooding of new information across the entire routing area after changes in any link state. With the growth of the network diameter or the frequency of link-state changes increases, the overhead in terms of bandwidth and processing cost, of flooding becomes prohibitive. Furthermore, such flooding over a large area will cause unnecessary overhead on the links, potentially creating many transient routing loops that can last for a long time. This limits the scalability of the routing protocols to large routing areas. To overcome such problems, we present in this paper an Optimized and Adaptive Link-State Protocol (OALP), a modification to the existing link-state routing protocol that does not require the state of each link to be flooded to the entire internetwork all the time, or to entire areas if we monitor the activity status of the nodes in the internetwork. Thus minimizing the amount of information distributed by link-state routing protocols.

There are primarily two modes in which we devise the network to operate depending the activity levels of the incumbent nodes. Depending on the activity levels an optimized flooding procedure is provided which would greatly reduce the number of advertisements flowing through the network.

1. Introduction:

There are two types of routing protocols Static Routing and Dynamic Routing protocol. Static routing is not really a protocol, simply the manual entry of routes into the routing table via a Routing, but it are manual and does not work well when it has to be entered on a large number of devices. It is also does not handle outages or down connections well, as the manual entries will have to be changed manually to recover from such a loss of connectivity. Distance Vector protocol is an example of Static routing protocol.

Dynamic routing protocols are software applications that dynamically discover destinations and how to get to them. A router will 'learn' routes to all directly connected networks first. It will then learn routes from other routers that run the same routing protocol. The router will then sort through its list of routes and select one or more 'best' routes for each destination it knows or has learned. Dynamic protocols will then distribute this 'best route' information to other routers running the same protocol, thereby extending the information on what networks exist and can be reached. This gives dynamic routing protocols to adapt to network topology changes or outages 'on the fly'.

In a link state protocol, like OSPF, every router has complete topology information about the network. OSPF routes traffic on shortest paths based on the advertised link weights. As a result, the links along the shortest paths may become congested while other links on longer paths remain idle. OSPF also allows for Equal Cost MultiPath (ECMP) where the traffic is distributed equally among various next hops of the equal cost paths between a source and a destination. This is useful in distributing the load to several shortest paths. However, as shown in, the splitting of load by ECMP is not optimal. The network traffic generated by the LSAs is enormous when there are lots of nodes, which are not part of the routing. Unnecessary

LSAs will be flowing throughout the network, which may not be used immediately by the nodes within that network. To alleviate this serious drawback, we propose a new strategy, which is optimized and adaptive routing protocol (OALP). The major goal of this framework is to minimize the number of LSAs flowing through the network at any point of time by using a threshold function previously determined. To realize this we suggest a simple perceptron model, which switches between the suggested framework and the conventional OSPF protocol depending on the traffic load.

A *topological database* is essentially an overall picture of networks in relationship to routers. The topological database contains the collection of LSAs received from all routers in the same area. Because routers within the same area share the same information, they have identical topological databases. The term *domain* sometimes is used to describe a portion of the network in which all routers have identical topological databases. Domain is frequently used interchangeably with AS. An area's topology is invisible to entities outside the area. By keeping area topologies separate, OSPF passes less routing traffic than it would if the AS were not partitioned.

Area partitioning creates two different types of OSPF routing, depending on whether the source and destination is in the same or different areas. Intra-area routing occurs when the source and destinations are in the same area; interarea routing occurs when they are in different areas.

An OSPF *backbone* is responsible for distributing routing information between areas. It consists of all area border routers, networks not wholly contained in any area, and their attached routers.

2. Suggested Framework for OALP

The suggested framework for OALP primarily consists of three phases, firstly the evaluation of a threshold value, secondly the simple preceptron model for switching and finally the Methodology for flooding.

2.1 Evaluation of the Threshold Value

The threshold value for the network is determined from the plot between load and number of LSAs per second (NLPS) for both the conditions. The minimum of the NLPS values is inspected to determine the load value for which switching from one form of OSPF to the other takes place.

Parameters in the measurement of the network load – number of LSAs/ second, network load.

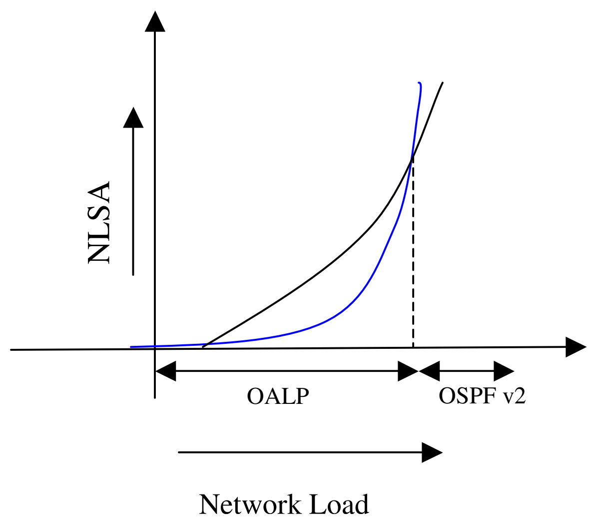

Fig 2.1.

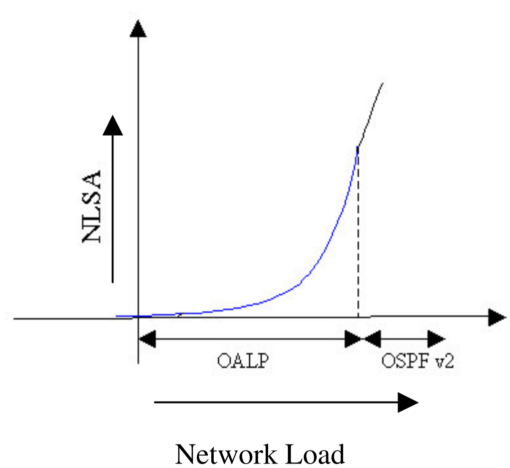

Fig 2.2

Fig. 2.1. LSA flood responses in case of reliable network scenario (the solid black line represents the normal OSPF protocol, the blue line represents the results when OALP is deployed for similar network loads)

Fig. 2.2. Final response got by superposing the responses of both protocols

In the figure above, response to increasing load is plotted against the number of link state advertisements per second. This goes to show the rapid increase in the number of advertisements with marginal increase in load after the

2.2 The perceptron model for switching

A simple perceptron model is defined which takes the weighted sum of all the activity corresponding to every node in the network. If the function value or the sum is higher than the threshold value decided by the training of the network, we switch to the older version of OSPF V2. In case the neural network does not fire, we employ the OALP modification to the OSPF.

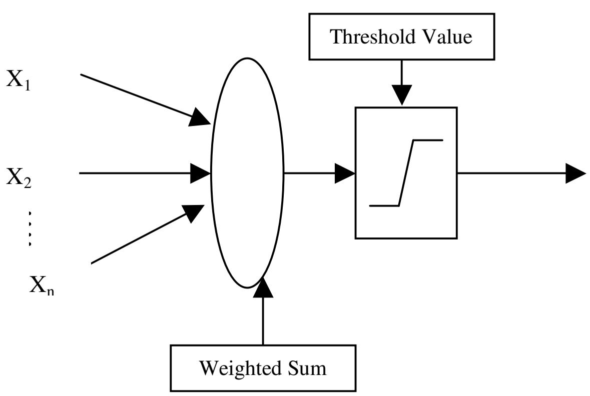

Fig 3. A simple perceptron model for the OALP

2.3 Methodology for flooding

This Method can be divided into three parts

1. Determining the Activity status.
2. The Activity Status Transition
3. The Flooding Procedure.

2.3.1 Determining the Activity status.

The OSPF protocol suffers from a major drawback in that it uses the flooding procedure to distribute the link state advertisements. Thus, many LSAs will be flowing in the network. To overcome this we provide an optimized flooding procedure. This Algorithm uses the Activity Status of a node to determine whether the LSA needs to be sent or not. The Activity Status of a node is determined based on the packets it has received or transmitted during certain interval of time. This algorithm restricts the flow of the LSAs to those nodes whose Activity Status is above par. Suppose a node is not involved too much in the packet transmission, repeatedly sending the LSA updates to such nodes would be too much of an overhead. Hence we restrict the flow of these LSAs to the nodes, which are actively involved in the packet transmission.

2.3.2 The Activity Status Transition

The following state diagram would depict the transition of the Activity Status of the node. The node can be two states:

a) The Inactive state (DOWN)

b) The Active state (UP)

Initially, the status of the node would be Up. Hence it receives the LSAs from all the other nodes. With this information it builds the Link State Database for itself. After that if its not involved in the packet transmission/receiving from any nodes during certain fixed interval of time, the status of the node reaches down state.
However, when a node needs to send a packet to other nodes, it sends a unique packet known as Active packet to every active node in the network. This prompts all nodes to send the LS Update to this node. The Link State Database is built on these LS Update Packets sent by every node. Here, the inconsistencies which might be seen during this process is removed by not allowing to send a node with a old Link State information which is determined by the timestamp which is maintained in every node which depicts the time stamp of the last LS Update packet it has received. Hence, in this case the old node, which has an old LS Database, is updated thereby making its state UP.

This procedure of maintaining the activity status of all nodes should be done within the network management system. This procedure reduces the unnecessary LSAs that are flowing in the network. Also, this procedure doesn't add any overhead apart from the Active packets, which flow only when it needs to get involved in the packet transmission. However this overhead would be completely overhauled by the optimization brought about by this flooding procedure.

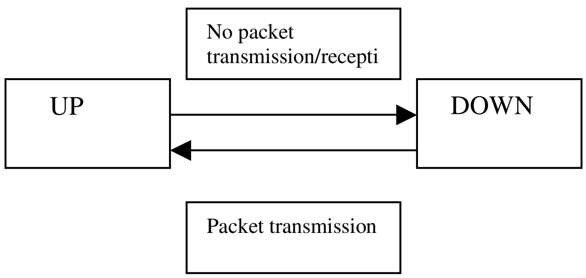

Fig 4. State transition diagram

2.3.3 Flooding Procedure

The Flooding procedure will be slightly revised to accommodate for this algorithm. The packets will be flooding to those nodes whose Activity Status is Up. This restricts the number of nodes it has to flow through. All nodes would see the activity status of its neighboring nodes and sends the LS Update Packets only if the status is Up. Also on receiving the Active Packet from any node, all nodes would send the LS Update Packets to the node requesting the Update. This completely restricts the flow of the packets throughout the network.

This Algorithm is mainly used in networks where there are lots of nodes, which are not participating in the packet transmission/reception. It will be serious overhead in networks where most of the nodes are actively participating. To overcome this we would like to provide a slight modification to the network management system. The network management system determines whether to apply this algorithm depending upon the activity of the nodes. If there is too much activity, our algorithm is not used and if there is lesser activity in the nodes our algorithm can be used to optimize the flooding procedure.

3. Results:

One of the major issues in determination of the threshold values and statistical values is to monitor OSPF routing. The method used here was to snoop into the LSAs already present in the network. We listen to the LSAs throughout the network by interacting with OSPF by creating stub routers. Though deployment is difficult the changes are no frequent than the network changes. So the underlying asynchronous theory is again used to reduce the overhead.

An estimate of the Number of LSAs and the overall traffic per unit time is then made. The load is the difference between overall traffic and the number of LSAs. This is plotted and the thresholds are determined. It has been observed that in a topology consisting of nodes more than 150, the number of LSAs in the network decrease by 22% than the previous values for similar values of network load prior to the threshold limits. While the results are positive if the number of nodes are larger, its observed that if the number of nodes are less than 34 there are no marked changes in the overhead.

4. Conclusions and Future Scope:

In this paper we have introduced a novel approach to reduce the unnecessary flooding of LSAs in a network, which runs OSPF as the routing protocol. Our approach is based on a statistical determination of the relationship between the load characteristics and the NLSA response. A switching technique is formulated based on this response and the network works is adjusted to a condition of the least number of LSAs flowing at a particular time.

We present a protocol based on a non-synchronous solution and its application to various traffic load scenarios. The correctness of this algorithm also can be seen by the route that is generated by this algorithm which is always the same as the route established by the OSPFv2. Also, this algorithm doesn't add to any overhead in case of large activity taking place in the network, as it is adaptive to the network traffic at that point of time.

The scope for future work in this area would be in trying to determine the scant resource approach to the shifting of one algorithm or another.

References

[1] JOHN.T.MOY, “OSPF RFC 2328” ASCEND COMMUNICATIONS, INC.APRIL 1998.
[2] C.Hedrick, ”RIP rfc 1058” Rutgers University, June 1998.
[3] John.T.Moy, “OSPF Complete Implementation” Sep 28, 2000; Published by Addison Wesley Professional.
[4] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[5] J. Moy, "OSPF: Anatomy of an Internet Routing Protocol," Addison-Wesley,1998.
[6] R. Guerin, A. Orda and D. Williams, "QoS routing mechanisms and OSPF extensions," *Proc. of Global Internet (Globecomm),* Nov. 1997.
[7] Moy, J., "OSPF Version 2", RFC 1583, March 1994.
[8] Moy, J., "Multicast Extensions to OSPF", RFC 1584, March 1994.