

EFFICIENT LSA FLOODING PROCESS TO OPTIMIZE OSPF ROUTING

P. Anu and S.Vimala

Department of Computer Science, Mother Teresa Women's University, India
anugps12@gmail.com, vimalaharini@gmail.com

ABSTRACT

In this paper, the Open Shortest Path First (OSPF) Routing algorithm is improvised in terms of congestion, packet delivery ratio and throughput by having the concept of cluster head to reduce the extra overhead that occurs due to LSA flooding.

Key words: Routing, OSPF, LSA, Cluster

INTRODUCTION

In recent years tendency of using internet as communication infrastructure for different telecommunication applications is growing significantly (Yuichiro, et al., 2007, Muhammad et al., 2005). In the current Scenario, Open Shortest Path First (OSPF) is one of the popularly used routing protocols of Xipeng et al., [1999] and Turgay, et al., [2002]. It is an observed fact that OSPF protocol may lead to increase in traffic engineering because OSPF uses shortest path always to forward packets. It will not reroute the packets in a less congested route of Franck, et al., [2014]. AS we all know, OSPF is the most popularly used link state routing protocol. In this, the complete network topology should be maintained by all routers. Through, the Periodic Hello Messages, OSPF is aware of its neighborhood. All, routers maintain Link state database (LSDB) and synchronizes it with neighbor's LSDB. Link State Advertisements (LSA) informs the topology information. Router sends the LSA to all the nodes except to the source through which it receives it. Thus, the Flooding of LSA takes place (Chandra, 2011, Sheela, 2010).

The reliability of this flooding process is achieved by sending LSA again until it gets an acknowledgement (Vetriselvan et al., 2014, Kyriakos Manousakis, et al., 2005). A topology change may result in generation of LSA of Anand Jayakumar, et al., 2017). After receiving new LSA updates their LSDB and recalculates the routing table to reflect new topology (Priyanka, et al., 2017).

LINK STATE ALGORITHM: Link state algorithm maintains the network topology. Dijkstra's shortest path algorithm is used to find the route and it is updated in routing table. It is forwarded to all nodes using Link State Advertisement (LSA). The flooding algorithm works as mentioned below:

1. Log is checked when a node receives LSA to check whether it is received already from the same interface.
2. If it is so then that LSA is discarded.
3. If it is new then it retransmits to all interfaces except the source.
4. Link state database (LSDB) is updated by the router.
5. The router calculated the shortest path of each and every node using Dijkstra's algorithm and it update its routing table. The algorithm finds the shortest path of each node hence the name.

The algorithm is described below:

1. The shortest path is evaluated by the node by considering itself as the root of the tree.
2. The Candidate list is filled by all its neighboring node.
3. From the above-mentioned list, the node which contains shortest path from root is the next child in the tree.
4. Step 2, Step 3 and 4 is repeated until node exists. If its distance from root only is taken and other routes are discarded.

CLUSTER FORMATION

Each and every OSPF area is a cluster. In cluster, nodes can be classified into two. Boundary nodes are nodes which are in the boundary of two clusters and have to pass information to both the clusters. The non boundary nodes have to passively listen to their cluster heads.

For efficient flooding, some nodes are assigned with special functions.

Flooding messages are forwarded only by the cluster heads to other nodes in the cluster.

It is the responsibility of the boundary nodes to forward the flooding packets to the neighboring Cluster Heads.

CLUSTER HEAD SELECTION

In our proposed algorithm each and every cluster should have its own cluster head. The change in topology is informed only to cluster heads by the nodes through LSA. All the cluster heads will in turn inform this information to the remaining cluster heads. The alternate path can only be finalized by the cluster heads in case of router failure. This leads to reduction in convergence time. LSA flooding among the nodes also reduced and in turn decrease the congestion.

This clustering algorithm enforces the idea: neighboring nodes are not considered as Cluster Heads. Except the cluster head all the nodes are connected to Cluster Head. The proposed algorithm determines the cluster head in a cluster:

Consider a 1 to m nodes b=1 to m-1 nodes

$$d_i = HL_i - DL_i$$

$$s_i = \sum d_{ij}$$

$$K(S) = \text{minimum}(s_i)$$

HL- Hierarchical Length in routing path(hops)

DL-Dijkstra's Length in routing path

In order to reduce the workload, the neighbor of the cluster head in a cluster sometimes acts as cluster head. Border router and cluster heads plays an important role if the source and destination belong to different clusters.

HL (Source, Destination) = HL (Source, Cluster Head1) + d (Cluster Head1, Cluster Head2) + HL (Destination, Cluster Head2)

HL (Source, Destination): Hierarchical distance from sender to receiver in terms of routing path

HL (Source, Cluster Head): Hierarchical distance from sender to Centre head of cluster1 in terms of routing path
 d (Cluster Head1, Cluster Head2): distance is calculated between the 2 Cluster heads of the respective clusters [12]

HL (Destination, Cluster Head2): Hierarchical length from destination to Centre head of cluster2 in terms of routing path [9]

Interaction between Cluster Heads and Router LSA Interaction from Center Head

Begin Within the cluster LSA is sent to all routers
 LSA is sent to other cluster heads

End

LSA Received at Center Head

Begin

Topology is updated in routing table

End

Router Sends LSA

Begin

Center Head receives LSA from router

End

LSA is received at Router

Begin

Topology is updated in routing table
 Dijkstra's algorithm is applied to find the shortest path. If there is any change in topology, then it is forwarded to its Centre Head through LSA

End

Traffic and congestion is reduced by using this proposed algorithm

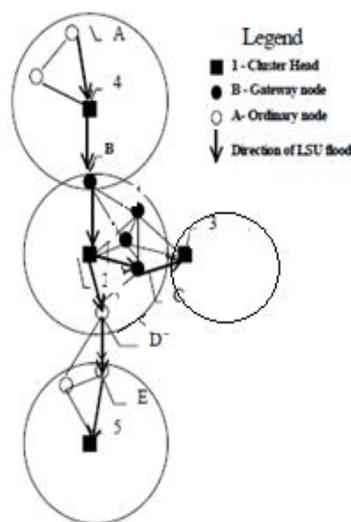


Figure 1: Cluster

The proposed algorithm is explained through the figure 1. All the clusters can be connected through gateway nodes among them. Let us consider, Node C as

the gateway to cluster 3, Node D as the cluster head for cluster 5 and B for cluster 2.

FLOODING

In this paper we describe our algorithm for the above-mentioned network. The gateway nodes are B,C and D. The algorithm is described as follows:

1. Node A sends all its Link State Advertisements only to its Cluster Head, node 4 but not to other nodes.
2. Cluster Head 4 updates the neighbor connectivity and broadcasts to all the nodes in the same cluster.
3. Since Node B is a gateway it sends it to neighboring Cluster Head 2.
4. Cluster Head 2 updates the neighbor connectivity and broadcasts to all the nodes in the same cluster.
5. Since Node C is a gateway it sends it to neighboring Cluster Head 3. Since Node D is a gateway it sends it to neighboring Cluster Head 5.
6. Cluster Head 3 updates the neighbor connectivity and broadcasts to all the nodes in the same cluster.
7. Cluster Head 5 updates the neighbor connectivity and broadcasts to all the nodes in the same cluster.

RESULT AND ANALYSIS

Packet delivery ratio is calculated as Received Packets by destination /forwarded packets by source* 100. This ratio is calculated for normal OSPF and for the proposed algorithm. From the below figure we can identify the proposed algorithm is always more.

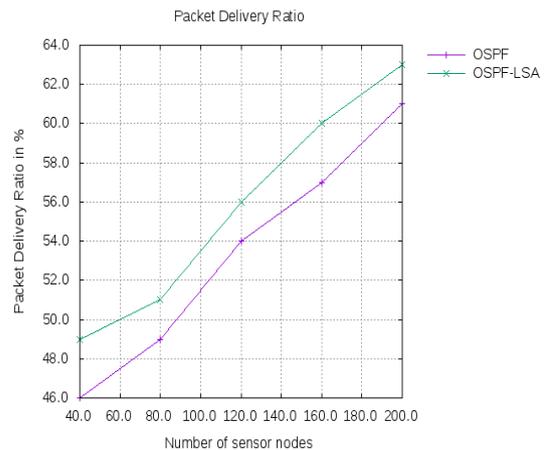


Figure2: Packet Delivery Ratio

End to end delay is calculated from the formula Received time of acknowledgement – sent time of message. For both OSPF and the proposed algorithm the above formula is applied and we can infer that the end to end delay is reduced by our proposed algorithm

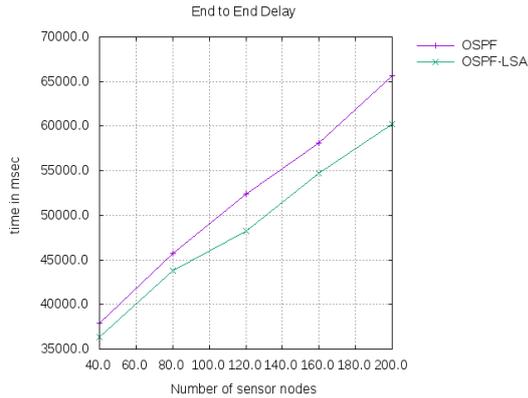


Figure 3: End to End Delay

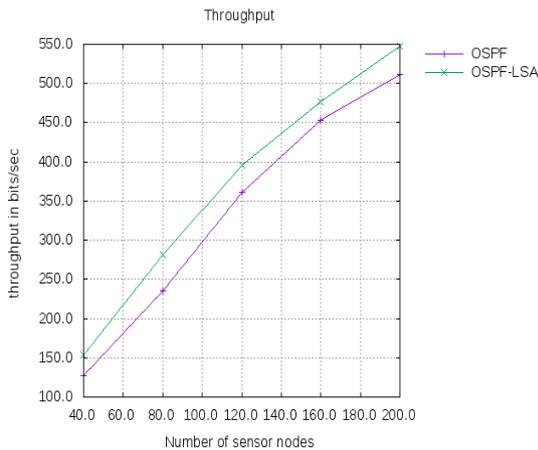


Figure 4: Throughput

Throughput is calculated as number of successful forwarded packet/time. This formula is applied for both the OSPF and proposed algorithm and we can find the throughput is increased in our proposed algorithm

CONCLUSION

In this paper, we proposed an efficient flooding algorithm to reduce LSA flooding through cluster concept. We performed simulation to evaluate the performance in terms of End to End Delay, Throughput and packet Delivery Ratio for normal OSPF and for our proposed system. It is observed that the proposed system outperforms OSPF with respect to the above-mentioned metrics.

REFERENCES

- Yuichiro H., T. Ogishi, S/ Ano and T. Hasegawa, KDDI R&D Laboratories, Japan, OSPF Failure Identification based on LSA Flooding Ana-lysis, IEEE (2007)
- Muhammad A., S. Javed, A.A. Chaudhry, M.H. Zaidi, W. Mahmood and A.A. Iqbal, NUST Institute of Information Technology, A Comparative analysis of Routing Protocols in GMPLS Pakistan, IEEE (2005)
- Xipeng X. and L. Ni, Reducing Routing Table Computation Cost in OSPF, IEEE (1999)
- Turgay K. and M. Krunz, Hybrid Flooding and Tree-based Broadcasting for Reliable and efficient Link-State Dissemination, IEEE (2002)
- Franck Le and J.L. Sobrinho, Interconnecting Routing Instances, IEEE/ACM Transactions on Networking, (2014)
- Chandra W., Indonesia, Performance Analysis of Dynamic Routing Protocol EIGRP and OSPF in IPV4 and IPv6 Network, 1st International Conference on Informatics and Computational Intelligence (2011)
- Sheela G.T., India 2nd International Conference on Computer and Network Technology, Dynamic Routing Protocol implementation decision between EIGRP, OSPF and RIP based on Technical Background Using OPNET Modeler (2010)
- Kyriakos M., V.T. Mc Auley and R. Morerav, J. Baras, International Conference on Wireless Networks, Communications and Mobile Computing, Using Multiobjective Domain Optimization for Routing in Hierarchical Networks (2005)
- Anand J.A., Krishnaraj C. and A.K. Nachimuthu, Aggregate Production Planning: mixed strategy. Pak. J. Biotechnol. 14 (3) 487-490 (2017)
- S. Priyanka, S., D. Shri and S. Poovizhi, Experimental investigation on partial replacement of steel slag as coarse aggregate. Pak. J. Biotechnol. 14(3): 481-485 (2017)