International Journal of Engineering Sciences Paradigms and Researches (IJESPR)
(Vol. 21, Issue 01) and (Publishing Month: June 2015)
(An Indexed, Referred and Impact Factor Journal)
ISSN (Online): 2319-6564
www.ijesonline.com

Implementation of OSPF Protocol for Directed and Undirected Graph

Preeti Sharma[1] and Trilok Gaba[2]

[1]M.Tech. Scholar, B.I.T.S., Bhiwani, Haryana (India)
preetitayal0@gmail.com

[2]Assistant Professor, B.I.T.S., Bhiwani, Haryana (India)
trilok_gaba@yahoo.in

Abstract

The OSPF is an open standard protocol that is most popularly used in modern networks. It is a link state protocol. Various protocols are used for shortest path. But in real life mostly problems are undirected graph like nature. OSPF using Dijkstra's algorithm solved shortest path problem in both type of problems i.e. directed and undirected graph. Along with problems becomes more crucial when no. of undirected branches are more than no. of nodes (eg. in road map, electrical circuits, protein structure, Genetic Engineering etc.).This research wills find how OSPF routing protocols works for directed and undirected graph problem. Following are the reasons of using OSPF: 1) OSPF is not a CISCO proprietary protocol like EIGRP.2) OSPF always determines the loop free routes.3) If any changes occur in the network it updates fast.4) Low bandwidth utilization.5) Support multiple routes for a single destination network.6) OSPF is based on cost of the interface.

Finally, use of Dijkstra's algorithm with sparse matrix will produce a running time of $O((|E|+|V|) \log |V|)$ instead of O(|V|^2 + |E|). Where V is the no. of nodes and E are the branches in graph.

Keywords: *OSPF, Dijkstra's algorithm, Directed and Undirected graph, Sparse Matrix, MATLAB.*

1. Introduction

OSPF is shortened form of Open Shortest Path First [1]. It is a dynamic routing protocol used in Internet Protocol networks. Specifically, it is a link-state routing protocol and falls into the group of interior gateway protocols, operating within a single Autonomous system. . OSPF was designed to support Variable-length subnet masking (VLSM) or Classless Inter-Domain Routing (CIDR) addressing models [2]. OSPF detects changes in the topology, such as link failures, very quickly and converges on a new loop-free routing structure within seconds. There are two types of routing-Link State routing and Distance Vector routing [3]. Dijkstra's is based on Link State routing. In Link State routing each router keeps track of its incident links and cost on the link, whether the link is up or down. Each router broadcasts the link state to give every router a complete view of the graph. Each router runs Dijkstra's algorithm to compute the shortest paths and construct the forwarding table. The topology of the network can be generated by collecting the OSPF messages.

2. Literature Survey

OSPF (Open Shortest Path First) and EIGRP (Enhanced Interior Gateway Protocol) are routing protocol which is a member of IGP (Interior Gateway Protocol). OSPF and EIGRP will distribute routing information between routers in the same autonomous system. This research wills find how OSPF routing protocols works[4].

Optimal traffic engineering (or optimal multi commodity flow) can be realized using just link-state routing protocols with hop-by-hop forwarding. Today's typical versions of these protocols, Open Shortest Path First(OSPF) and Intermediate System-Intermediate System (ISIS),split traffic evenly over shortest paths based on link weights [5]. However, optimizing the link weights for OSPF/ISIS to the offered traffic is a well-known-hard problem and even the best setting of the weights can deviate significantly from an optimal distribution of the traffic.

As the RIP protocol and OSPF protocol are the main router protocols used in the network. OSPF was designed to support Variable-length subnet masking (VLSM) or Classless Inter-Domain Routing (CIDR) addressing models. OSPF detects changes in the topology, such as link failures, very quickly and converges on a new loop-free routing structure within seconds. It computes the shortest path tree for each route using a method based on Dijkstra's Algorithm [6], a shortest path first algorithm

The OSPF is an open standard protocol that is most popularly used in modern networks. It is a link state protocol. It features the concept of areas to provide scalability. The key factor in designing an OSPF network is the assignment of router and its links to an area(s), which is whether it has to been put in Area 0 (Backbone) or any other non-backbone area [7].

Another protocol, EIGRP (Enhanced Interior Gateway Routing Protocol) features a unique diffusing update algorithm (DUAL) [8] for calculating best paths to a destination. It is extremely powerful in reducing the convergence delay that we typically have in a modern network. EIGRP also has a great quality of being very easy on CPU utilization for devices. It is scalable; it does accommodate very large networks. EIGRP features a very simple configuration. Automatic summarization is enabled by default; so EIGRP acts in a class full manner and automatically summarizes prefixes. The advantages using EIGRP are as follow [9]:

1) Easy to configure.2) Loop free routes.3) Keeps backup path to the destination network.4) Convergence time is low and bandwidth utilization.5) Support Variable Length Subnet Mask (VLSM) and Classless Inter Domain Routing (CIDR).6) Supports authentication.

The disadvantage of using EIGRP is as follow [10]:

1) Considered as Cisco proprietary routing protocol.2) Routers from other vendor are not able to utilize EIGRP.

3. Methodology and Purposed Scheme

Dijkstra's Algorithm

Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. The algorithm exists in many variants; Dijkstra's original variant found the shortest path between two nodes, but a more common variant fixes a single node as the "source" node and finds shortest paths from the source to all other nodes in the graph, producing a shortest path tree [11,16].

For a given source node in the graph, the algorithm finds the shortest path between that node and every other. It can also be used for finding the shortest paths from a single node to a single destination node by stopping the algorithm once the shortest path to the destination node has been determined [17]. For example, if the nodes of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities [18]. As a result, the shortest path algorithm is widely used in network routing protocols, most notably IS-IS and Open Shortest Path First (OSPF). It is also employed as a subroutine in other algorithms such as Johnson's. Dijkstra's original algorithm does not use a min-priority queue and runs in time O(|V|^2) (where |V| is the number of nodes. This is asymptotically the fastest known single-source shortest-path algorithm for arbitrary directed graphs with unbounded non-negative weights. In some fields, artificial intelligence in particular, Dijkstra's algorithm or a variant of it is known as uniform-cost search and formulated as an instance of the more general idea of best-first search [19,20].

Algorithm

Let the node at which we are starting be called the *initial node*. Let the *distance of node Y* be the distance from the *initial node* to *Y*. Dijkstra's algorithm will assign some initial distance values and will try to improve them step by step.

1. Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes.
2. Set the initial node as current. Mark all other nodes unvisited. Create a set of all the unvisited nodes called the *unvisited set*.
3. For the current node, consider all of its unvisited neighbors and calculate their *tentative* distances. Compare the newly calculated *tentative* distance to the current assigned value and assign the smaller one. For example, if the current node *A* is marked with a distance of 6, and the edge connecting it with a neighbor *B* has length 2, then the distance to *B* (through *A*) will be 6 + 2 = 8. If B was previously marked with a distance greater than 8 then change it to 8. Otherwise, keep the current value.

4. When we are done considering all of the neighbors of the current node, mark the current node as visited and remove it from the *unvisited set*. A visited node will never be checked again.
5. If the destination node has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the *unvisited set* is infinity (when planning a complete traversal; occurs when there is no connection between the initial node and remaining unvisited nodes), then stop. The algorithm has finished.
6. Otherwise, select the unvisited node that is marked with the smallest tentative distance, set it as the new "current node", and go back to step 3.

Purposed Scheme Set up Parameters:

Table 4.1 Shows the values of various set up parameters used for simulation purpose in our thesis.

(Set Up Parameters)

Set up parameter	**Value**
Tool Used	MATLAB 2013a (8.0)
Nodes position	Random
Number of nodes& Edges	6 nodes and 11 edges
Routing algorithm	Dijkstra's Shortest Path
Graph Used	Directed & Undirected
Mobility Model	Random Walk
Routing Protocol used	OSPF

Weight values for Directed and Undirected Graph:
DG =

Nodes	(6,2)	(1,6)	(2,3)	(5,3)	(3,4)	(4,1)	(4,6)	(2,5)	(5,4)	(6,3)	(1,5)
Weight	.41	.99	.51	.32	.15	.45	.38	.32	.36	.29	.21

UG =

Nodes	(4,1)	(5,1)	(6,1)	(3,2)	(5,2)	(6,2)	(4,3)	(5,3)	(6,3)	(5,4)	(6,4)
Weight	.45	.21	.99	.51	.32	.41	.15	.32	.29	.36	.38

4. Simulation and Results

Tool Used- MATLAB

MATLAB is a high performance language for technical computing. It integrates computation, visualization and programming in and easy to use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB is an interactive system while basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix a vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C. the name MATLAB is an abbreviation of matrix laboratory [21,22]. MATLAB features a family of add-on application specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulations and many others. The MATLAB system consists of five main pans, when MATLAB is started the MATLAB desktop appears, containing tools for managing files , variables and applications associated with MATLAB [23].

Simulink

Simulink is a software package for modeling, simulating and analyzing dynamic systems. It supports linear and non-linear systems, modeled in continuous time, sampled time, or a hybrid of the two. Systems can also be multirate i.e. have different parts that are sampled or updated at different rates [24,25]. For modeling, Simulink provides a graphical user interface (GUI) for building models as block diagrams, using click and drag mouse operations. It includes a comprehensive block library of sinks, sources, linear and non-linear components and connectors. You can also customize and create your own blocks. For information on creating your own blocks, like separate writing S-Functions. Models are hierarchical, so you can build models using both top down and bottom up approaches. You can view the system at a high level and then double click blocks to go down thorough the levels to see increasing level of model detail. This approach provides insight into how a model organized and how its parts interact.

The model can be simulated using a choice of integration methods, either from the Simulink menus or by entering commands in the MATLAB Command window. The menus are particularly convenient of interactive work, while the command line approach is very useful for running a batch of simulations (for example using scopes and other display blocks, you can see the simulation results while the simulation is running, the simulation results can be put in the MATLAB workspace for post processing and visualization [26].

Snap shot & Simulation Results

Snap shot for various numbers of nodes have been taken, as shown by the figures.

- Fig. 4.2 shows snap shot for 6 nodes and 11 edges for directed graph.
- Fig. 4.3 shows snap shot for result of shortest path 6 nodes and 11 edges for directed graph.
- Fig. 4.4 shows snap shot for 6 nodes and 11 edges for undirected graph.
- Fig. 4.5 shows snap shot for result of shortest path 6 nodes and 11 edges for undirected graph.

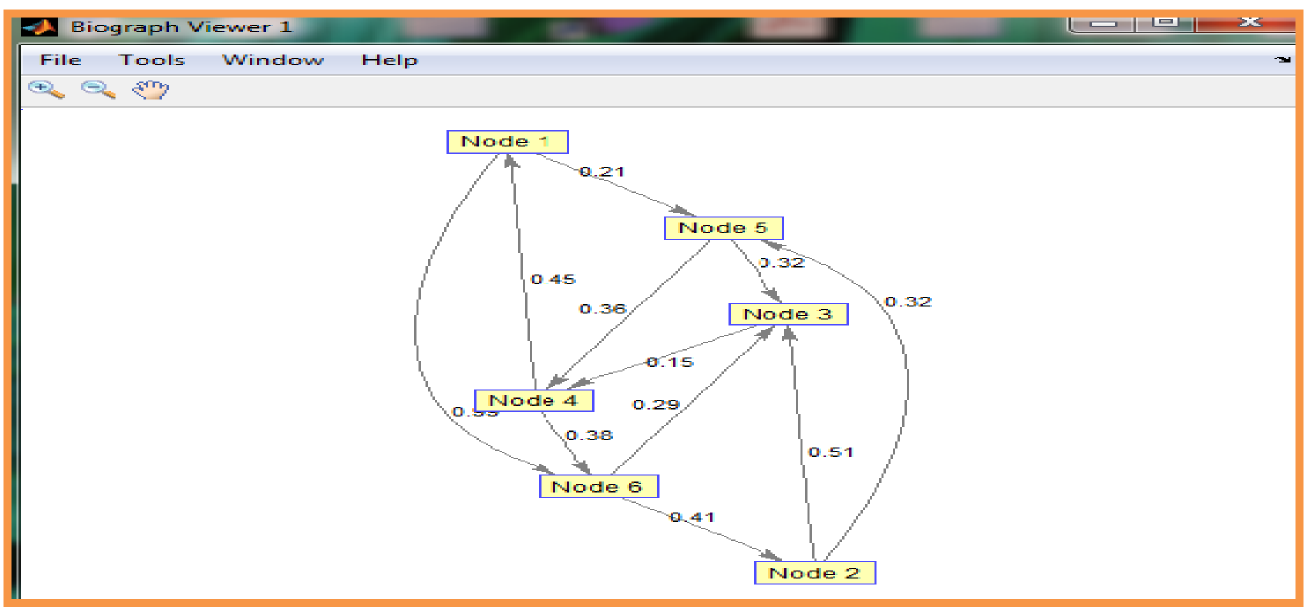

Fig. 4.2 shows snap shot for 6 nodes and 11 edges for directed graph.

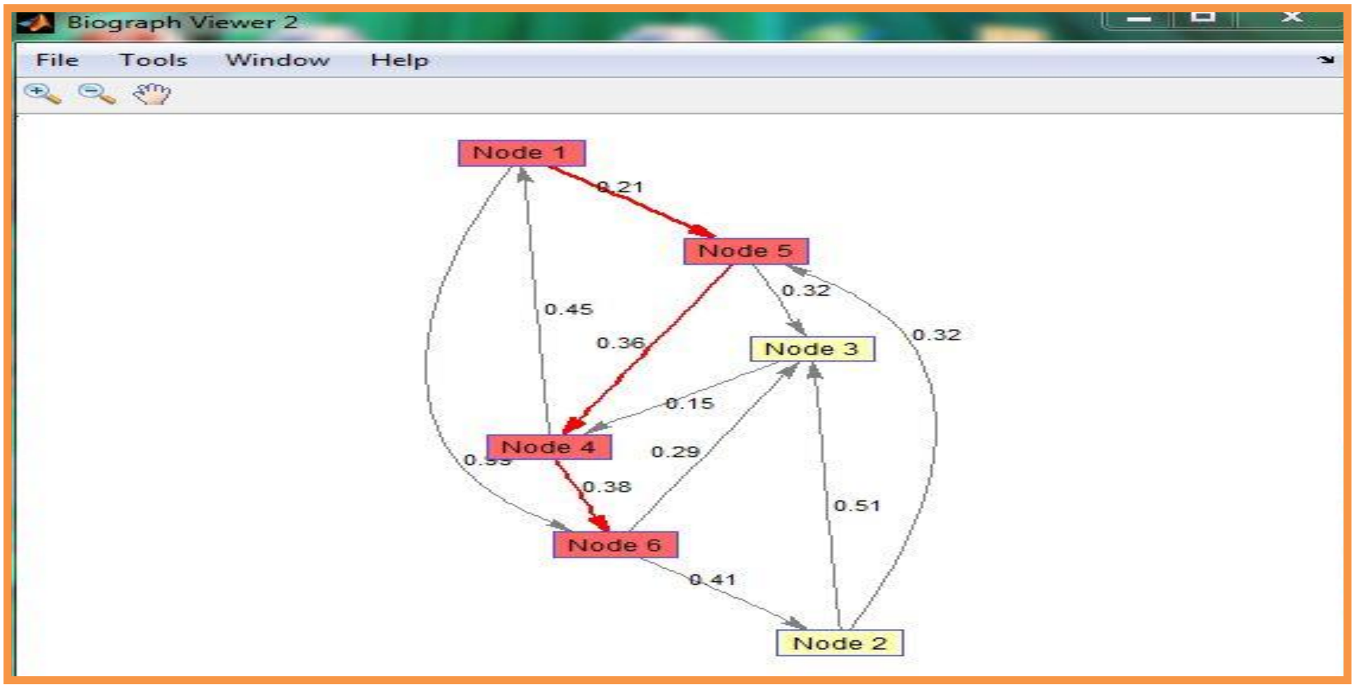

Fig. 4.3 shows snap shot for result of shortest path 6 nodes and 11 edges for directed graph.

Shortest path in the graph from node 1 to node 6: 1 5 4 6 (Distance: .9500)

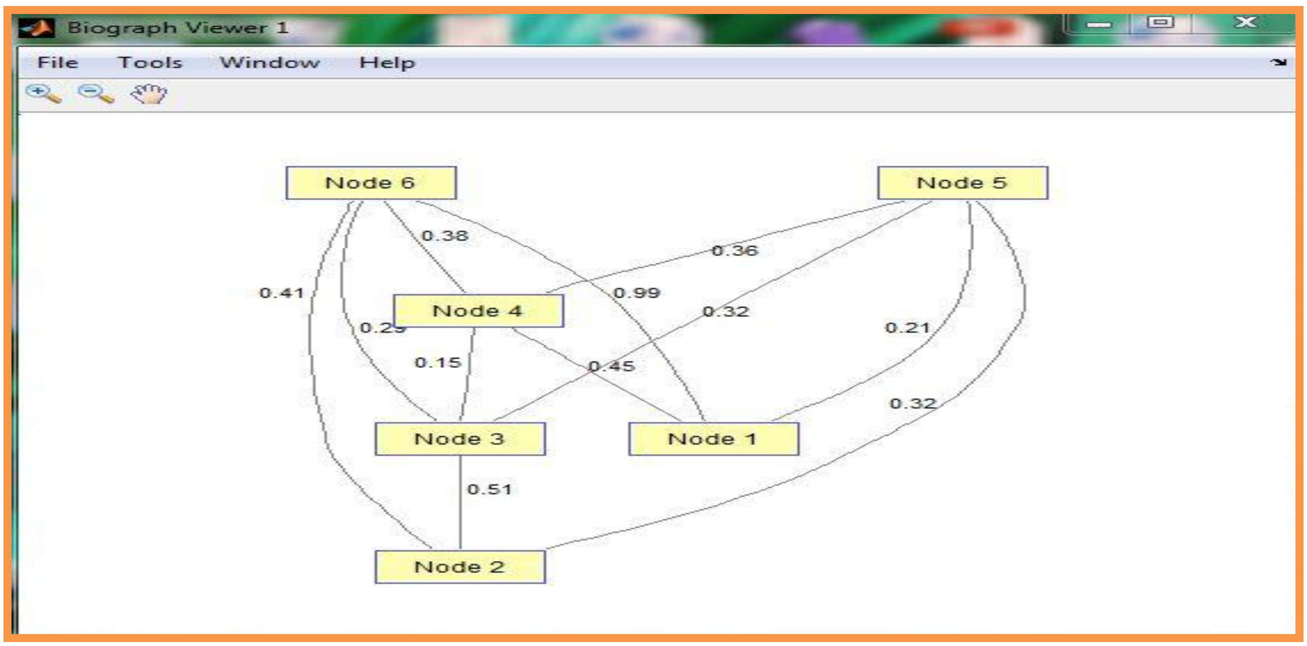

Fig. 4.4 shows snap shot for 6 nodes and 11 edges for undirected graph.

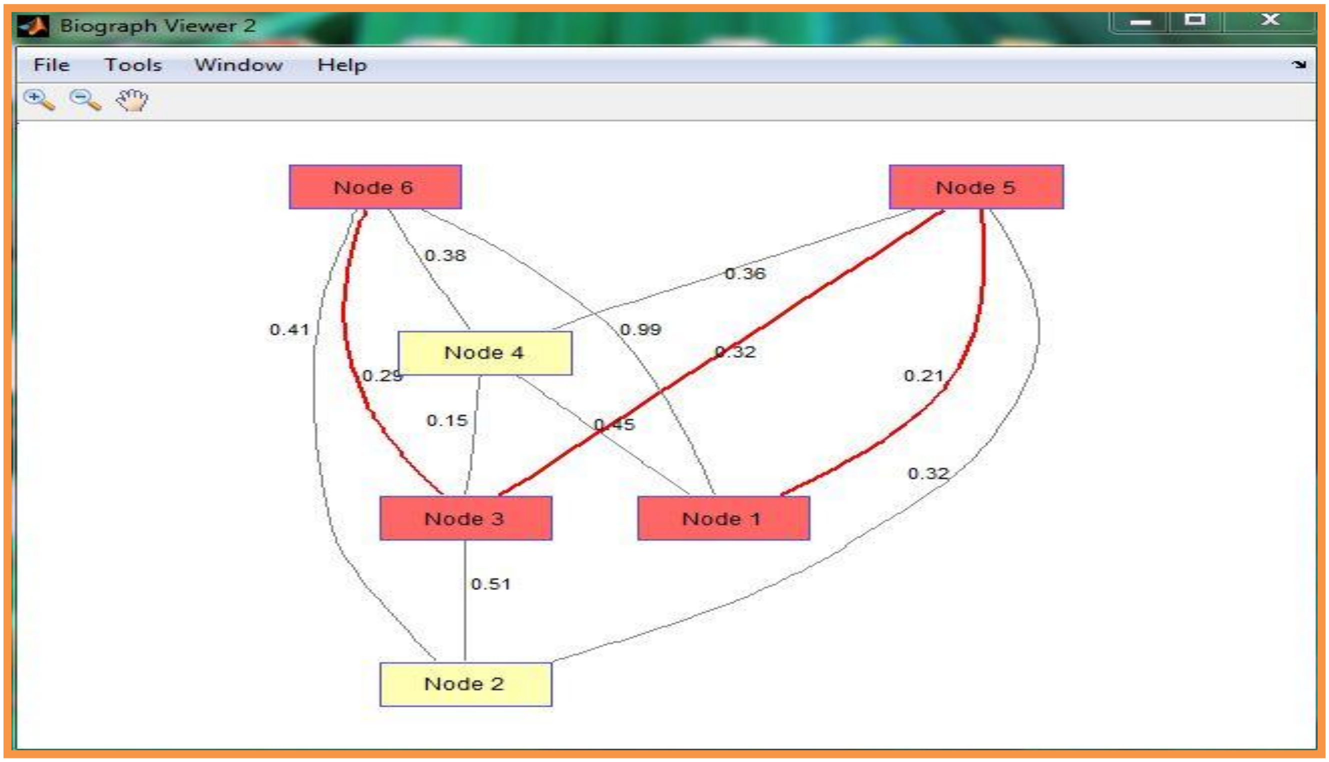

Fig. 4.5 shows snap shot for result of shortest path 6 nodes and 11 edges for undirected graph.

Shortest path in the graph from node 1 to node 6: 1 5 3 6 (Distance: .8200)

5. Conclusion and Future Scope

Conclusion

Results for directed and undirected graphs are successfully implemented in MATLAB 2013a with desired results. The OFDM implementation is to store vertices in an array or linked list will produce a running time of O(|V|^2 + |E|). For sparse graphs (with very few edges and many nodes), it can be implemented more efficiently storing the graph in an adjacency. This will produce a running time of *O*((|*E*|+|*V*|) log |*V*|).

Finally, we made sure that it is a correct algorithm (e.g., it *always* returns the right solution if it is given correct input. With the help of two mathematical results:1) Triangle inequality If δ(u,v) is the shortest path length between u and v, δ(u,v) ≤ δ(u,x) + δ(x,v). 2) The sub-path of any shortest path is itself a shortest path. We can claim that any time we put a new vertex in network, we can say that we already know the shortest path to it.

Future Scope

1. Traffic Information Systems are most prominent use
2. Mapping (Map Quest, Google Maps), where multiple ways are available for same destination. (i.e. low nodes high edges)

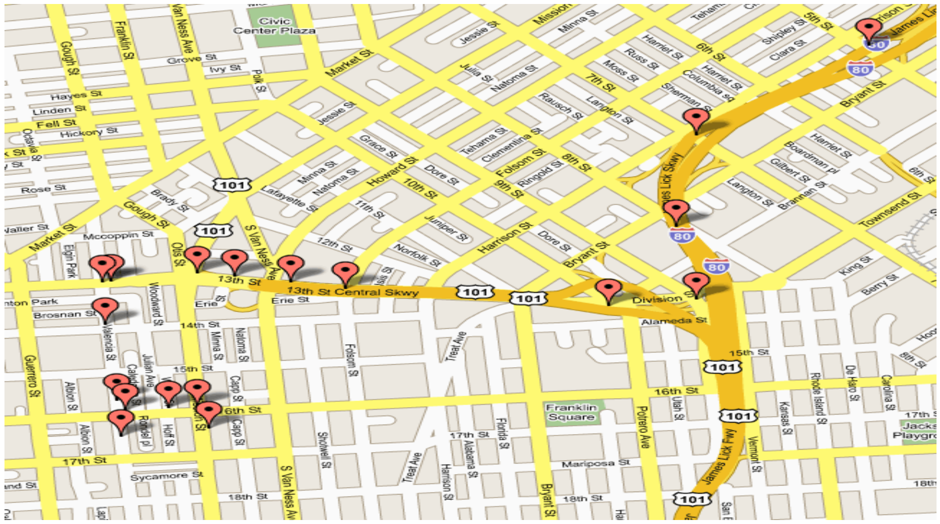

Figure 5.1

3. Epidemiology: to model the spread of infectious diseases and design prevention and response strategies. Vertices represent individuals and edge their possible contacts. It is useful to calculate how a particular individual is connected to others. Knowing the shortest path lengths to other individuals can be a relevant indicator of the potential of a particular individual to infect others.

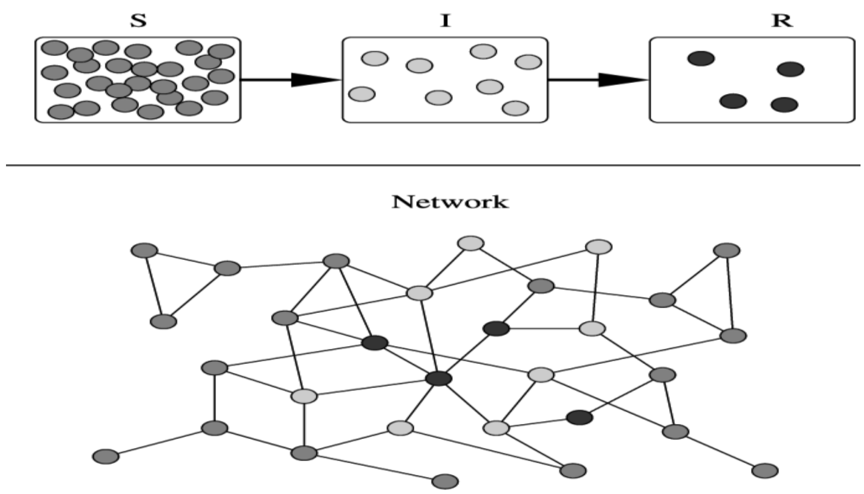

Figure 5.2

4. Wireless extensions of OSPF to support mobile ad hoc networking, with major focus on design and implementation of one of the most promising proposals
5. IPv6 Addresses with Embedded IPv4 Addresses using OSPF protocol.
6. Genetic interaction network

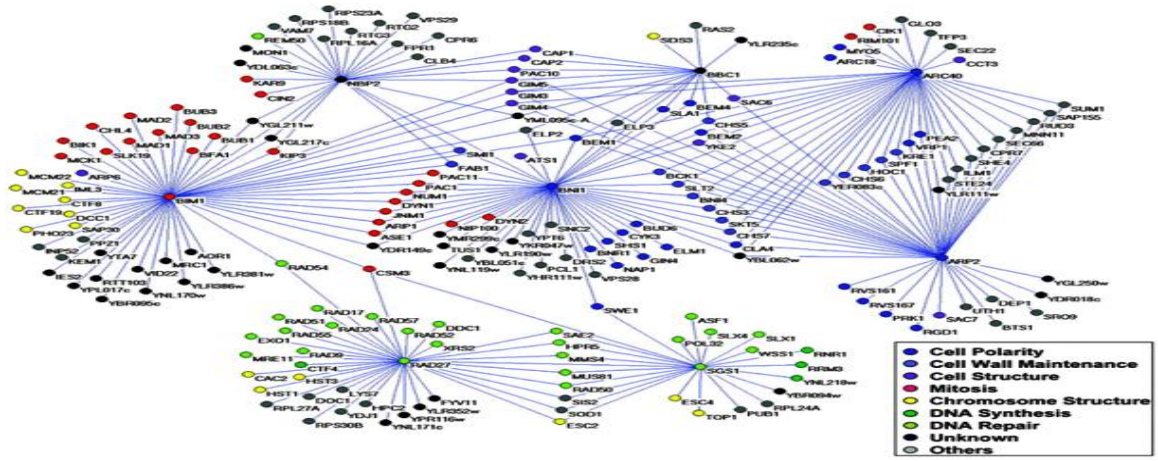

Figure 5.3

References

[1] Chandra Wijaya "Performance Analysis of Dynamic Routing Protocol EIGRP and OSPF in IPv4 and IPv6 Network", 2011 First International Conference on Informatics and Computational Intelligence.

[2] Chris K. Williams "Tuning Dynamically Routed Internet Protocol Networks to Achieve Controlled and Predictable Failover during Link Instability".

[3] Talal M. Jaafar, George F. Riley, Dheeraj Reddy "Simulation-Based Routing Protocol Performance Analysis", Proceedings of the 2006 Winter Simulation Conference.

[4] Sheela Ganesh Thorenoor"Dynamic Routing Protocol Implementation Decision Between EIGRP, OSPF And RIP Based On Technical Background Using OPNET Modeler", Second International Conference on Computer and Network Technology.

[5] AlirezaSarikhani, MehranMahramian, HamidrezaHoseini"Calculation of Cisco Router Processing Power for a Large Network with Thousands of Nodes", 2010 2nd International Conference on Signal Processing Systems (ICSPS).

[6] DahaiXu, Member, IEEE, Mung Chiang, Senior Member, IEEE, and Jennifer Rexford, Senior Member, IEEE, Fellow, ACM"Link-State Routing With Hop-by-Hop Forwarding Can Achieve Optimal Traffic Engineering", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 19, NO. 6, DECEMBER 2011.

[7] John Dwyer, HetalJasani"An Analysis of Convergence Delay Caused By Link Failures in Autonomous Systems".

[8] CosminAdomnicai, ViorelMinzu "Method forAdjacency Information Updating In Ipv6 Networks".

[9] Antonio Cianfrani, Vincenzo Eramo, Marco Listanti, Marco Marazza, Enrico Vittorini"An Energy Saving Routing Algorithm For a Green OSPF Protocol", This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings.

[10] IoanFitigau, Gavril Toderean"Network Performance Evaluation for RIP, OSPF and EIGRP Routing Protocols", IEEE, 2013.

[11] G. P. SaiKalyan, D.VenkataVara Prasad "Optimal Selection of Dynamic Routing Protocol with Real Time Case Studies", IEEE, 2012.

[12] Bezruk Valery, VarichVyacheslav"The Analysis of the Characteristics of Routing Protocols in IP Networks", TCSET'2010, February 23-27, 2010, Lviv-Slavske, Ukraine.

[13] Yao Zhao, Jian Liang Yan, HuaZou"Study On Network Topology Discovery in IP Network", Proceedings of IC-BNMT2010.

[14] Song —Yang, Zhang Zhi—Yong"RIP Internet Protocol Failure Analysis and Research" 2012

International Conference on Industrial Control and Electronics Engineering.

[15] Rajeev Rastogi, Yuri Breitbart, Minos Garofalakis, Associate Member, IEEE, and Amit Kumar "Optimal Configuration of OSPF Aggregates", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 11, NO. 2, APRIL 2003.

[16] PankajRakheja, Prabhjotkaur, Anjali gupta, Aditi Sharma "Performance Analysis of RIP, OSPF, IGRP and EIGRP Routing Protocols in a Network" International Journal of Computer Applications, June 2012.

[17] Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs"

[18] Mehlhorn, Kurt; Sanders, Peter (2008). Algorithms and Data Structures: The Basic Toolbox. Springer.

[19] Felner, Ariel (2011). Position Paper: Dijkstra's Algorithm versus Uniform Cost Search or a Case Against Dijkstra's Algorithm

[20] R. C. Prim: Shortest connection networks and some generalizations. In: Bell System Technical Journal, 36 (1957), pp. 1389–1401.

[21] Russell, Stuart; Norvig, Peter (2009) [1995]. Artificial Intelligence: A Modern Approach (3rd ed.). Prentice Hall.pp. 75, 81.

[22] Sometimes also least-cost-first search: Nau, Dana S. (1983). "Expert computer systems" (PDF). Computer (IEEE) 16 (2): 63–85.

[23] Sniedovich, M. (2010). Dynamic Programming: Foundations and Principles. Francis & Taylor.

[24] horup, Mikkel (2000). "On RAM priority Queues".*SIAM Journal on Computing***30** (1): 86–109.

[25] Thorup, Mikkel (1999). "Undirected single-source shortest paths with positive integer weights in linear time".journal of the ACM 46 (3): 362–394.

[26] OSPF Working Group Mailing List Archive, ftp:// gated.cornell.edu/pub/lists/ospfJohn Moy, OSPF Version 2, RFC 1583, March 1994

[27] M. Cooper and D. C. Yen. "IPv6: business applications and implementation concerns," Computer Standards and Interfaces, vol. 28, no. 1, pp. 27–41. July 2005.

[28] X. Wen, C. Xu, J. Guan, W. Su, and H. Zhang, "Performance investigation of IPsec protocol over IPv6 network," in Proc. Of Artificial Intelligence Applications & Innovations, Larnaca, Cyprus, October 6-7, 2010, pp. 174-177.

[29] Y. Lu, W. Wang, Y. Zhong, and B. Bhargrava "Study of distance vector routing protocols for mobile ad hoc networks," in Proc. of First IEEE International Conference on Pervasive Computing and Communications, 2003, pp. 187-194.

[30] V. J. Wang, J. Yang, G. Xie, and M. Zhou, "OSPFv3 protocol simulation with colored Petri nets," in Proc. of International Conference on Communication Technology, Beijing, China. April 09-11, 2003, pp. 247-254.

[31] N, Ayub, F. Jan, T. Mustafa, W. J. Rana, M. Y. Saeed, and S. Ullah, "Performance analysis of OSPF and EIGRP routing protocols with respect to the convergence," European Journal of Scientific Research, vol. 61, no. 3, 2011, pp. 434-447.